WORDQUEST
A COMPUTER ASSISTED INSTRUCTION SYSTEM FOR THE DRILLING OF ENGLISH GRAMMAR

A CREATIVE PROJECT

SUBMITTED TO THE GRADUATE EDUCATIONAL POLICIES COUNCIL

IN PARTIAL FULFILLMENT OF THE REQUIRMENTS

for the degree

MASTER OF ARTS

by

ZAHYDEE G. MINNICK

Committee Approval:

5/23/83
Committee Chairman
Date

5/23/83
Committee Member
Date

Elizabeth M. Riddle
5/23/83
Committee Member
Date

Department Head Approval:

5/23/83
Head of Department
Date

Graduate Office Check:

David L. Wheeler
5/23/83
Dean of Graduate School
Date

BALL STATE UNIVERSITY

MUNCIE, INDIANA

JULY, 1983

CREATIVE PROJECT ABSTRACT

CREATIVE PROJECT: Wordquest. A Computer Assisted Instruction System for the Drilling of English Grammar.

Student: Zahydee G. Minnick

Degree: Master of Arts

Date: June, 1983

Pages: 114

This project concerned the development of a computer software system to help students reinforce material previously discussed in the classroom.

The software is written in Apple Pilot, a computer language especially designed to help teachers with little or no programming background develop simple computer assisted instruction courseware. The program creates an adventure style game where the student finds treasure, fights monsters, and overcomes obstacles by answering questions related to English comparatives and superlatives.

WORDQUEST
A COMPUTER ASSISTED INSTRUCTION SYSTEM
FOR THE DRILLING OF ENGLISH GRAMMAR

A CREATIVE PROJECT
SUBMITTED TO THE GRADUATE EDUCATIONAL POLICIES COUNCIL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
for the degree
MASTER OF ARTS
by
ZAHYDEE G. MINNICK
ADVISER - HERBERT STAHLKE

BALL STATE UNIVERSITY
MUNCIE, INDIANA
JUNE, 1983

Thesis
LD
2489
.Z72
1983
.M56

TABLE OF CONTENTS

1. Introduction.................................... 1

2. Description of the Project.
2.1 Instructional Level............................. 4
2.2 Programming Level..............................10
2.2.1 Wordquest: Declaration of Variables......11
2.2.2 Map1: Top Flow Control...................12
2.2.2.1 Map Labels.......................14
2.2.2.2 Quit and Exit Routines..........15
2.2.2.3 Cave Routines....................16
2.2.3 Monsters1: Control of Questions.........18
2.2.4 Question File............................21
2.2.5 Summary and Reviews......................23
2.2.6 Question Counter.........................24
2.2.7 Animation................................25

3. Instructions for the User.
3.1 Running the Program............................27
3.2 Implementing New Material......................30

4. Compendium of Programming Techniques for Pilot Users.
4.1 The Editor.....................................33
4.2 Accessing Files with the Apple Pascal Language System...............................35
4.3 Text-Files.....................................37
4.4 Graphics Design................................39
4.4.1 Text-Windows.............................40
4.4.2 Quick and Slow Draws.....................41
4.4.3 Recalling Graphics.......................41

5. Project Evaluation..............................44

References and Bibliography........................46

Appendices:

I. Program Listings
  A. Wordquest......................................49
  B. Map1...56
  C. Monsters1......................................66
  D. Sumup..74
  E. Review...80
  F. Question.......................................90
  G. Skedemo..99
  H. Checkpoint.....................................101

II. Flow-Charts
  A. Wordquest. General Flow-Chart..................104
  B. Monster Routine................................105
  C. Cave Routine...................................106

III. Graphic Printouts.
  A. Cave Entrance..................................108
  B. Map1...108
  C. Skeleton.......................................109
  D. Dragona..109
  E. Cerberus.......................................110
  F. Warlock..110
  G. Cliff..111
  H. Ruins..111
  I. Book...112
  J. Exit...112

IV. Skeleton. Sequence 1, Frame 1.......................114

1. INTRODUCTION

This project concerned the development of a computer software system for instructional purposes. The use of computers in education has increased considerably in the past years as educators have started to realize that they can be powerful teaching tools.

> Films, television and slides are valuable because as educational tools they can help bring into the classroom experiences which the teachers can not. (Poirot. 1980:9).

The same can be said about computers especially now that the cost particularly of micro-computer systems is within the reach of many schools.

The use of computers in teaching is commonly known as Computer Assisted Instruction (CAI). According to Poirot (1980) there are three major types of CAI system: drill and practice, simulation, and tutorial. In the first case the purpose of the program is to reinforce material which has already been taught in the classroom. This is particularly helpful when teaching languages, math, or any other subject that requires drilling.

Simulation programs are very useful in science-related classes since all kinds of experiments can be performed in a simulated form using computers. Tutorial systems are used to present material to the student and have him or her answer questions concerning the information. In this case it is not assumed that the student has prior knowledge of the subject. He can cover the material at his own pace and progress through it in conjunction with or independent-ly from the classroom teacher.

It is very important to clarify that this project was concerned with the development of a drill and practice system as opposed to a tutorial or simulation system. As was mentioned above, this type of system is designed to reinforce material previously discussed in the classroom.

In general, this drill and practice system can be used with any kind of teaching material, from English to science or math, according to the needs of the teacher. Nevertheless, it is specifically implemented to cover English comparatives and Superlatives. Instructions to modify or extend the topic material are discussed in section 3.2 which covers specific information about implementation of new material.

A general review of the different degrees of comparison for adjectives and adverbs is presented. Rules to deal with positive, comparative and superlative constructions properly are explained, drilled and tested. The level of the material is appropriate for high school students, grades 9 thru 12, and for intermediate students of English as a Second Language.

Considering the fact that this system is used to review and drill classroom material it would be appropriate to assume that its place in a language institute or high school curriculum should be that of a laboratory period. Computers should be located in the language lab. This way when students attend their lab period, they would have the choice either to drill pronunciation material in the regular booths or to drill grammar structures in the computer. Thus, the lab would be serving several purposes, always oriented to meet the needs of the students.

The program is written in Apple Pilot, a computer language designed to help teachers with little or no computer programming background develop simple C.A.I. courseware. With a deeper knowledge of computer programming more complex programs can be developed. Information concerning the structure and use of the program is given in later chapters.

2. DESCRIPTION OF THE PROJECT

2.1 Instructional Level.

The material concerned with this project is presented through the medium of an adventure game. For motivational purposes a variety of carefully designed pictures is included as well as some simple music compositions. While wandering around the caves of Wordquest, the student finds monsters and obstacles which lead him either to review or question sets. It is very important to point out that given the design of the program and the capabilities of Pilot, the student can actually physically participate in the adventure by using the Apple game controls.

At the beginning of the program the student is asked his name and then given the choice either to skip the instructions section or to go ahead and read it, depending on whether or not he has run the program before. This is specially convenient for students who are not running the program for the first time since the instructions are rather long.

At this point the student is teleported to Level 1 of Wordquest and a map of the caves, (Appendix III-B), is displayed along with the cross-hairs which he can move around by using the Apple game controls. This gives the student control of the system so that he can choose his move of action.

By looking at the map we observe several asterisks, some labeled, others not. The labeled asterisks are designed to guide the student toward routines of particular importance such as 'Book', which displays a summary of the topic material to be covered; 'Information', which displays the score, the number of questions already answered, the number of questions left to answer, and the number of right and wrong answers given. It also displays the number of gold and life points obtained. By choosing 'Quit' the student is allowed to leave the program at any time he wants.

The unlabeled asterisks are meant to be the surprise element of the adventure. Two of them lead to cave rooms, where the reviews are to be found, and the remaining four lead to encounters with the monsters who are in charge of displaying the different sets of questions.

When getting to a cave room a picture is displayed along with the cross-hairs. At this point he has the alternative of choosing from three unmarked asterisks, two of which correspond to review sets, and one to a dummy routine, where he gains or looses life or gold points. By choosing the marked asterisk 'Exit' he is sent back to the map.

It is important to notice that when he gets a review, he has the choice of reading it right away or later in the program. If he chooses to read it, the review is diplayed and then he is automatically sent back to the map. If he chooses not to read it right away, a record is kept and he is given the choice of reading it before the assigned questions are displayed. As soon as he leaves the caves and returns to the map, the corresponding asterisk is labeled with the name of the cave to avoid giving him an unnecessary visit to a place already seen.

The material covered in each review set corresponds to two monster routines. In other words, when facing a specific monster his questions will be related to the material covered by those reviews. Since both monsters and caves correspond to unmarked asterisks, the student may easily find a monster before actually finding and reading the assigned review. In this case he may go back to the map and continue looking for the assigned review or go ahead and answer the questions anyway. In the first case the student is sent back to the map; in the second case, the question set assigned is displayed.

In both of the cases mentioned above, a record is kept and the asterisk is marked with the number of the question set just accessed. For example:

? set # 1

If the student has actually answered the questions the purpose of the label is to show him that those questions have already been answered, therefore he is not supposed to access them again. If he does, a message will be displayed and he will be sent back to the map. If he choses not to answer the question, the purpose of the label would be that of a marker, so that when he finds and reads the reviews, he can return to that particular set of questions.

Besides the 'Sum-up', which is a summary of the topic to be covered, there is a total of four review sets, each dealing with specific rules regarding the use of comparatives and superlatives. Review one deals with rules concerning monosyllabic and polysyllabic adverbs and adjectives, review two with disyllabic adjectives and adverbs, review three with irregular forms for the comparative and superlative, and review four with positive comparisons.

Each review is accompanied by a set of questions for the purpose of drilling the material rather than testing it. In this case since no score of right or wrong answers is being kept, the student may answer each question at least twice before the correct answer is displayed. He is then asked another question or sent back either to the cave or to a monster routine, depending on where he originally came from.

There is a total of four monster routines, each one containing a set of four questions, each set assigned to a review. For instance, the questions on set number one correpond to the material covered in review number 1. This could be called the actual testing part of the material. At this point, the student is permitted to answer each question only once and a score of right and wrong answers is kept and displayed for him to see his progress.

The format of the questions varies from true/false to sentence completion and multiple choice. A record of the number of questions answered and remaining is kept; thus when all questions have been answered the asterisk in the right bottom part of the map is labeled with the word EXIT. This means the adventure is over and he is free to leave. Since it is rather time-consuming to go through the whole adventure, a QUIT option is included which allows him to leave the program at his convenience.

There are several advantages to drilling and reviewing material in this way. First, the idea of the adventure is highly motivating, especially since the student is presented with neatly designed graphics along with the text material. It is important to emphasize that the graphics were designed to fit the idea of the adventure through the dungeons and not to fit the review material. In other words the student will find graphics of caves and monsters instead of pictures which would illustrate the grammatical patterns taught.

In contrast with many other languages, Apple Pilot has the capability of letting text appear within the graphics, which makes them more useful. The fact that the student can actually manipulate the cross-hairs by using the game controls adds a lot of movement to the adventure.

Another important advantage to consider is the fact that the student can go through the material at his own pace. A 'Press Return to Continue' routine is included after every picture or full-text screen. This allows him to stop and read the material without having to deal with a screen full of rolling lines. By pressing the return key the next screen of information is displayed. In addition the student is given immediate feedback, since a right or wrong answer comment is always displayed as soon as he types in an answer.

One of the disadvantages of using this kind of system is the fact that if such a system is not carefully designed, the student might go just for the amusement of the game. This is something rather difficult to control in this particular program, since it is designed to give the student the option of reading the reviews if he thinks he is not prepared enough, or avoiding them when he thinks that he knows the material well enough. To a certain extent this is remedied when dealing with the question sets. At that point the student is not given the choice to answer or not. He is required to answer and if he skips the reviews and is not prepared enough it is very likely that he will do poorly throughout the entire adventure.

2.2 Programming Level:

The program is basically composed of eight text files, seven graphic files, one character set file, and one data file, all of them being stored in the diskettes called TEXT1 and GRAPH1.

The text-files contain the following information:

Wordquest: Introductory explanation and declaration of variables. (Appendix I-A).

Monsters1: Routines used to call the graphics to be displayed, setting of text windows for these graphics, comments of the monsters and control of the questions to be used. Appendix (I-C).

Sumup: A summary of the topic material to be drilled. Appendix (I-D).

Review: A set of reviews regarding the topic material. Appendix (I-E).

Question: The questions to be displayed along with the right and wrong answer routines. (Appendix I-F).

Skedemo: The set of instructions required to animate the character set Eskeleto (Appendix I-G) and, Map1: Instructions that control the program flow. Appendix (I-B).

The graphics were designed to fit the idea of the adventure; thus, there are four graphics of monsters, a dragon, a three-headed dog, a skeleton and a warlock. There is also a graphic of the entrance to the cave and two cave rooms, one with a cliff and the other with some deserted old ruins. A graphic of a book was designed to be displayed right before accessing the summary and reviews, and a graphic of the exit from the caves is also included.

The figure of a tiny skeleton is animated and used to end the program. It was created in the character set editor by assigning parts of it to different letters of the alphabet. The parts are saved under the file named SKELETON and are assembled in a data file called MUEVELO. The text file Animatrix contains the instructions used to actually animate the different sequences.

2.2.1 Wordquest: Declaration of Variables.

In Pilot it is not necessary to declare variables. Simple numeric variables are created simply by using them in a compute instruction. For instance:

```
C: G=10
```

String and numeric array variables must be dimensioned and initialized at the beginning of the program. For instance:

```
D:Name$(30)       *String Variable*

D:G(4)            *Dimension Numerical Array*

C:G(1)=0
C:G(2)=0          *Initializing elements*
C:G(3)=0          *of the array*
C:G(4)=0
```

For the purpose of organization, all the variables used through the program are listed and initialized at the beginning. Due to a lack of disk space, not all the neccessary information regarding the function of these variables is included in the program listing. Nonetheless, further information about each specific variable is given when the contents of each file are explained in detail.

The structure of the program is clearly presented in the flow-charts included. (Appendices II-A, II-B, and II-C). The flow-control in each chart is centered around the A.P. (Accept Point instruction) which displays the cross-hairs in the screen giving the user the option to select the move that determines the instructions to be executed next.

2.2.2 Map1: Top Flow Control

After the proper instructions are given, a map of the caves is displayed along with the cross-hairs. In the map there are several labeled and unlabeled asterisks which correspond to particular routines. This map constitutes the top flow-control of the system.

When the map is displayed along with the cross-hairs, control of the system passes to the student who moves the intersecting point of the cross-hairs towards any of the asterisks in the map and then presses one of the game control buttons. The instruction:

```
AP:
```

displays the cross-hairs, and then depending on the coordinates of the intersecting point chosen, Pilot executes the next instruction.

By looking at the listing of the file Map1, the following instructions can be seen:

```
AP:
C:H=%X
C:V=%Y
```

The first instruction displays the cross-hairs, and then the value of the coordinate X is stored in the variable H and the value of the coordinate Y is stored in the variable V. From then on, every instruction regarding the coordinates of any of the asterisks must be made in terms of H and V. For instance:

```
L(h>186 and h<194 and v>479 and v<487):inform
```

In this case if the coordinates of the asterisk chosen correspond to the ones mentioned above, the program jumps to the label Inform as stated in the instruction. Specific instructions to define the coordinates X and Y are given in the printout of the file Checkpoint. (Appendix 1H).

2.2.2.1 Map Labels.

Although the map displays only three labeled asterisks, the program was designed to label the remaining ones after their assigned routines have been accessed. To do this, two numeric arrays were dimensioned and set to 0 at the beginning of the program:

```
D:(O)2 and D:P(4)
```

The elements P(1), P(2), P(3) and P(4) correspond to each one of the monster routines while the elements O(1) and O(2) correspond to each one of the cave routines. When the cave and/or monster routines are accessed the assigned elements are updated to 1. The process of labeling the asterisks is accomplished by the following set of instructions:

```
J(P(1)=0): Label2
G:M44,237
G:T?Set #1

*Label2
```

If the monster routine assigned to the element P(1) has been accessed, the value of P(1) would be 1, thus the instructions to move the cursor to the coordinates 44,237 and type ?Set #1 would be executed, but if that specific routine has not been accessed yet the value of P(1) would continue to be 0, therefore the program would jump to label 2, according to the instructions.

Every time the map is displayed, the program checks on every routine that has been accessed and automatically labels it. It is important to point out that this process slows the the program considerably since it must check all the options before displaying the cross-hairs for the student to make his next move.

2.2.2.2 Quit and Exit Routines.

Other routines included in the file Map1 which should be explained further are the Quit and Exit . Since the program is rather long, the student is given the choice of quitting at any time. On the other hand, if he goes through the whole adventure, an Exit label is displayed to let him know that the adventure is over. The bottom-left asterisk is labeled Exit only when the value of the variable Q (questions) is 0. If he happens to choose this asterisk before all the questions are answered, a message is displayed and he is returned to the map.

Both of these options use a routine called Inform, which displays the number of questions remaining, and the number of gold and life points obtained. This routine can also be accessed at any time the student moves the cross-hairs toward the asterisk in the map labeled Information. At the end of the routine Inform, the instructions:

```
J(E=1):Print
J(E=0):Map
```

determine whether to jump to the label Print which displays a message and links to the file Skedemo to end the program or to the map to continue with the adventure. If the options Quit or Exit are chosen, the variable E is updated to 1, thus the instructions to end the program are executed. If the option Information is chosen, the value of the variable E remains 0, therefore the program jumps back to the map and the student can continue with the adventure after getting the wanted information.

2.2.2.3 Cave Routines.

Depending on the coordinate points accessed, the program either links to other file-labels or jumps to a label within the same file. If the asterisks assigned to the cave routines are accessed, the program jumps to the labels Ruins and or Cliff which contain the instructions to call the graphics from the diskette Graph1, set text-windows for them and control the access to the different sets of reviews.

In each cave routine, three unlabeled asterisks in addition to a labeled one are displayed. Two of the labeled asterisks correspond to review sets and the third one to a dummy routine where gold and life points are lost or obtained. The asterisk labeled Exit allows the student to return to the map.

The elements of the numeric array O which are initially set up to 0, are updated to 1 as soon as a cave routine is accessed. The instructions:

C:O(1)=1 and or C:O(2)=1

are used to tell the program to label the asterisks assigned to the cave routines.

When a hint assigned to a review is found, several variables are updated:

C:R(1)=1: Tells the program that review one has been found, thus if the student does not want to read it right away, he can choose to do so later in the program before the questions assigned to this review are displayed.

C:G=G+5: Updates the value of gold points. Generally the student is suppossed to pay in gold for each review he finds and wants to read.

C:H=0: This tells the program that the review is accessed from the caves, therefore after displaying the review, the program should return to the caves.

After the student decides to read the review, the file is linked to the Review file with the instruction:

LY:Review,one

where Y means link if the prior match instruction was yes, Review is the file to be linked to and One is the specific label that must be accessed. If the student does not want to read the review immediately, the instruction:

```
JN:Busca
```

jumps back to the beginning of the cave routine where the cross-hairs are displayed.

At the beginning of all the text-files a Press Return to continue routine is included, to allow the student the reading of a full-screen at a time. The instruction:

```
U:Return
```

jumps to the label Return and executes the following instructions before returning to the place where the routine was originally called from.

```
T:Press Return to Continue. (Displays this message).
W:2                         (Waits for two seconds).
A:                          (Accepts the response).
G:ES                        (Clears the screen).
E:                          (Ends the routine).
```

2.2.3 Monsters1: Control of Questions

If the coordinates assigned to one of the monster routines are accessed, the program links to the file Monsters1 where the instructions to call the graphics, to set the text-windows and to control the questions are stored. The instruction:

```
L(h>14 and H<22 and v>239 and v<247): Monsters1,Eskele
```

links to the file Monsters1, specifically the label Eskele. There, the elements of the array P are updated to 1, for instance:

```
C:P(1)=1
```

which causes the program to label the asterisk assigned to the set of questions just answered before returning to the map.

The first time this routine is accessed, the value of the array element M(1) is equal to 0, which means that that particular set of questions has not been answered. It is updated to 1 when the Question file is accessed in order to display the assigned question set. The instruction:

```
J(M(1)=1):SK2
```

causes the program to jump to the label SK2 if the value of M(1) is equal to 1. At this point a message is displayed to tell the student that he has already answered that particular set of questions and since he is allowed to attempt to answer the questions only one time, the program links back to the map.

If the value of M(1) remains 0, the instructions that follow the command are executed. Some comments are displayed and the program jumps to either the label check1 or check2 by the instruction:

```
J(R(1)=0):CHECK1
J(R(1)=1):CHECK2
```

The value of R(1) is equal to 0 if the review assigned to that set of questions has not been accessed yet, if it has, the value of R(1) equals 1. In the first case the student is told that the reviews are not available for him to read so he is given the choice either to go ahead and answer the questions anyway or to go back to the map and continue his search for the review. The instruction:

```
LY:Question,One
```

links to the file Question, label One if the last match instruction was yes, while

```
LN:Map1,Map
```

links back to the file Map1, label Map.

In Check2, the student is told that the reviews are available for him to read if he wants to do so. If he wants to do it the program links to the file Review, label One. If he prefers not to do so, the program links to the file Question, label One.

After the questions are displayed, the program links back to the file Monsters1, label Score. Depending on how well the student did, the program executes the next instruction:

```
J(C1=4):Win1
```

If the number of correct answers given in that particular set equals 4, the program jumps to Win1 where the student is congratulated for having answered all the questions in the set correctly. This means he has beaten the monsters. The idea of the defeated monster is obtained by reversing all the colors in the screen with the command

G:es8

After being informed about the number of life points he has gotten, the program links back to the map. If the number of correct answers given is not equal to 4, the instructions written next are executed. The student is informed about the number of answers he got right as well as the number of life points obtained. At this point no defeat takes place, so he does not get to see the destruction of the monster. After this the program links back to the map.

2.2.4 Question File.

The text-file Question contains the different set of questions to be displayed along with a right and wrong answer routine. Several variables are updated when accessing this file:

C:Q=Q-1

Every time the right or wrong answer routine is accessed, the value stored in the variable Q(question) is decreased by one. The value of Q being equal to 0 causes the program to write the label EXIT for the student to know that the lesson is over.

C:C=C+1

C stands for correct answer. The instruction increases the number of correct answers by 1 every time the routine is accessed.

C:W=W+1

Wrong stands for wrong answer. The number of wrong answers is increased by 1 every time the routine is accessed.

C:C1=C1+1

This instruction increases the number of right answers given in each specific set by 1. This variable is set to 0 at the beginning of each question set. It is used to keep the student informed of how well he did in each set. The score of the total amount of correct and incorrect answers given can be obtained at any time the student chooses the coordinates assigned to the label Information.

There is a total of four question sets, each containing four questions. Each question uses either the right or wrong answer routine depending on whether the last match instruction was correct or not. For instance:

Th:

Displays the question and leaves the cursor right after the question mark.

PR:U

Converts all the letters in the answer to capitals.

A:

Accepts the response.

M:

Attempts to match the response given.

UY: Right

If the answer is correct, it uses the right answer routine and returns.

UN:Wrong

If the answer was not correct, it uses the wrong answer routine and returns.

It is worth pointing out that the M(match) instruction is a very useful feature of Pilot. It allows the teacher the use of different question formats such as multiple choice, yes-no, information and completion of sentences. It allows the student to type in complete words or just the initials, as well as several answers at the time if it is necessary. At this point it is very important that the teacher be creative enough to predict the kind of answers the student might come up with because, even though the instruction is very effective, it will not match any thing the teacher has not programmed.

2.2.5 Summary and Reviews.

The content of the reviews is basically divided in two files, one Sumup, where general information about the material to be covered is given, and Review, which contains all the reviews to be displayed. The reviews were intended to be short and to the point. A couple of questions are included after each review with the intention to drill the material rather than to test it. Since no score is being kept at this moment, the student has the choice of answering each question twice before the right answer and comments are displayed.

2.2.6 Question Counter.

Another nice feature of Pilot is the possibility of using system variables, one of which is the question counter variable. This variable automatically stores the true or false value of an answer (true being 1 and false being 0) and increases it by one every time the same accept instruction is used. In other words, the student can answer the same question as many as nine times before the right answer is displayed or before jumping to another label. This is accomplished by the following instructions:

TY1: That was very well.

Displays the message if the student answers the question correctly the first time.

JY1: Next

Jumps to the next label if the answer given was correct.

T1: That was not very good, try again.

Displays the message if the answer given for the first time is incorrect.

T2: You are not doing very well, so here is another chance.

Displays the message if the answer is incorrect the second time through.

T3: The right answer is....

Displays the right answer if for the third time the answer given in incorrect.

J3: Next

Jumps to the next label.

J: Start

Jumps to the label start, in order to display the question again. After each review set is displayed, the program links either to the caves or to the Question file, depending on where the file was called from.

It is very important not to include any other A(accept) instruction while keeping track of the attempst to answer the question. Doing this would result in the question counter being set up to 0 and its purpose would not be achieved. This is why return to continue routines were used, while displaying the questions,since this routine uses an Accept instruction.

2.2.7 Animation.

In order to accomplish the animation of the tiny skeleton, several files were used. First, the character set file called Eskeleto, where the parts of the skeleton were drawn and assigned to the different letters of the alphabet. The data file Muevelo, created while running Animator, is used to assemble the parts of the skeleton in frames and sequences and the text file Skedemo actually animates the different sequences as desired. As noted before, a clear explanation of how this is done is not included in the manuals. Nevertheless, by looking at the Pilot program Maxwell, a general idea can be obtained. In the creation of the moving skeleton, most of the instructions were taken exactly as they appear in the listings of the mentioned program.

It is very important to notice that when assembling the different frames, dots must be included between each frame in order to avoid the overlapping of sequences when running the program. For instance:

SEQUENCE 1

Frame 1　　　　　　**Frame 2**

```
...ab..          ...ab..
..cdef.          ..cde=.
.ghijk.          ..hij-.
.l#nop.          ..mno<.
...qrs.          ..>qr^.
...tu..          ...AB..
...v...          ...CD..
...w...          ...EF..
...xO..          ...GH..
...yz..          ...IJ..
```

The letters and characters included in frame 1 correspond to the whole body of the skeleton in one position. (Appendix IV). Some of these letters reamin unchanged in the other frames since they are assigned to unmoving parts of the body, such as the trunk and the head. The other letters and characters are asigned to moving parts such as the arms and the legs. Each frame corresponds to a complete figure in a different position.

3. INSTRUCTIONS FOR THE USER

3.1 Running the Program:

Insert the diskette labeled TEXT1 into drive #1 and the diskette labeled GRAPH1 into drive #2. Turn the computer on, and when asked the name of the lesson to run, type in:

WORDQUEST

After a couple of minutes the program will start running.

Every time the message 'Press Return to Continue' is displayed, make sure to press the key labeled Return after the cursor has appeared in the screen. After a welcome to the program is shown, the question:

Have you run this program before?

will be displayed. If you have, type 'Yes' and you will be teleported to the caves. If you haven't, type 'No' and instructions to run the program will be displayed.

The map of the caves will be displayed along with the cross-hairs. Move the intersection of the cross-hairs toward the asterisk of your choice and press one of the paddle's buttons. Make sure the intersection is as centered as possible, otherwise you might miss the exact coordinates. If this happens, you will simply get a message telling you to try again.

As you will observe, there are three labeled asterisks within the map. These labels are used to refer you to routines of particular importance such as 'Book' where you can find a summary of the material covered in the reviews; 'Information', where the updated numbers of gold and life points, and questions left, and a score of right and wrong answers are displayed, and 'Quit', which allows you to stop execution of the program.

One of the unlabeled asterisks is assigned to a routine that displays the label 'Exit' as soon as all the questions have been answered. This is used to let you know that the adventure has been completed. Choosing it causes the program to be terminated.

Two unmarked asterisks correspond to cave routines where you will find gold or hints and gain or loose life points. When you find a hint you will be given the choice of reading it right away, but if you do not want to do so, you will be given the chance of reading it again right before the questions assigned to that review are displayed.

Four unmarked asterisks are assigned to monster routines, where, faced with the monsters, you will have to answer their questions to remain alive. If you meet a monster before having found the reviews, the monster will always give you the chance to back up. In this case you can continue your search for the reviews and later on come back.

After answering a set of questions, you will not have a chance to answer it again. If you try to do so, you will get a message and the program will automatically return you to the map.

To Yes-No questions, the safest way to answer is by typing in Yes or No depending on the answer. Nevertheless, answers such as ok, all right, sure, will sometimes be accepted. Multiple choice questions can be answered either by typing the number of the right answer or by spelling it out. For instance: 4 or four.

After each set of questions, the score of right and wrong answers given will be displayed to let you know how well you did in that particular set. If you want to see the total score obtained so far, refer to the asterisk in the map labeled 'Information'.

When returning to the map either from the caves or one of the monster routines, the assigned asterisk will be labeled automatically, for instance:

?Set #1 (Question set number 1)

In this way you can keep track of the places you have already been in, or you can use them as a point of reference if you need to go back.

3.2 Implementing New Material:

A general overview of the structure of the program can be easily obtained from the flow-charts and all the commands are fully explained in the Apple Pilot Tutorial. Nevertheless, neither provides any guide as to how the program could be modified or expanded. To do this the teacher does not need to have a deep knowledge of computer programming however, on the contrary, by following a brief set of instructions he or she will be able to modify or expand either the summary, reviews or the questions to fit his or her needs.

The development of different practice material can be done by editing three text presentation files: SUMUP, which is a summary of the material to be reviewed, REVIEW, which covers the reviews or hints the student finds while wandering through the caves and QUESTION, which includes the questions to be displayed. It is very important that these files remain under the given names, otherwise all the link commands would be useless and the program would not run.

These files are stored in the diskette called TEXT1 which also contains all the other files used by the program, therefore space availability is rather limited. For this reason, when modifying the program, the SUMUP, QUESTION and REVIEW files must be created in a separate diskette and then copied onto diskette TEXT1 after deleting the original files. These files are a component of the main program, therefore they include variables which are updated every time the files are accessed. For this reason it is very important that these files be created according to the format specified in the printouts of the mentioned files.

Since retyping most of the routines included in a separate file would be time consuming, it would be convenient for the teacher to copy the file or files he wants to change onto a new initialized diskette. Furthermore, he can edit the file or files, but he must make sure that only the object of the T(type) instructions are modified, since none of the instructions themselves can be changed. In other words, the content of the material can be modified but all the labels and instructions concerning the running of the program must remain the same.

After all the necessary changes have been made and the file has been saved, the teacher can delete the original file and copy the new version of it onto the diskette Text1. To prevent any irremediable loss, the teacher should make sure he has back ups of all the files before proceding to delete them.

It is also very important that the teacher be aware of the fact that each file must not exceed a total amount of 18 blocks otherwise, when copying them into the TEXT1 diskette, he or she would get a 'writing error' which is defined as 'inability to write file due to a lack of disk-space'. Fortunately, as was mentioned above, PILOT uses a very complete editor which will let the teacher know the number of blocks used and the number of blocks available each time he saves the program.

4. COMPENDIUM OF PROGRAMMING TECHNIQUES FOR PILOT USERS

A thorough analysis of Apple Pilot would be too extensive to cover, therefore only its applications concerned with this project will be discussed.

Apple Pilot was specifically designed for teachers with little or no programming background to help them develop simple C.A.I. programs. The system comes with the Author and Lesson diskettes besides the Apple Pilot Editor and Language Reference Manuals.

Both of the manuals are considerably easy to use. The instructions are very clear and are always illustrated through short programs. One of the qualities Pilot offers is an immense capability for graphic design and use, which, needless to say, plays a very important role in the motivational part of any teaching material. Simple music compositions and non-standard alphabets can also be created. With a deeper knowledge of the language, the teacher can develop animation. Nonetheless it is important to point out that the instructions to do this are both unclear and insufficient.

4.1 The Editor

The system includes a very effective editor, composed of several menus that illustrate the different options available. A general menu is displayed as soon as the system is started giving the user the choice to specifically access the text, graphic, sound and or character set editors. It also gives the option to initialize or duplicate diskettes. Specific instructions regarding the use of any of these editors are clearly explained in the Pilot Editor Manual.

The text of the lesson is created in the lesson-text editor, and saved on a file for later use. Graphics can be created in immediate or deferred mode, in other words, they can be created in the graphics editor by using the Apple paddles or by means of programming instructions within the lesson text. Simple music compositions can be created using the Sound editor or by means of programming instructions. The non-standard alphabets can be created only in immediate mode.

The language itself is very easy to learn since it consists of only-twenty two instructions.

> Each instruction consists of two main parts: the instruction itself and the object of the instruction. The instructions themselves are very simple, but many of them can operate on a wide variety of objects. Furthermore, you can use modifying elements with an instruction to change its operation and conditioning elements which determine whether or not the instructions will be executed under various conditions. (Apple. 1980c:2).

Information concerning the execution of instructions based on conditioning elements is given in the explanation of contents of the Map1 routine. Specific examples are given in the print out of the file Map1, under the label Try.

4.2 Accessing Files with the Apple Pascal Language System.

It is very important to differentiate between text-files and data-files. A text file is any file created in the text editor. It is a set of instructions stored under a file name, usually referred to as Lessons, in the editor manual. Any text file automatically appears in the directory. A data file is a file created within a file according to specific instructions. Data files do not appear in the directory, which is rather inconvenient. Nonetheless, data files can be accessed by using the Apple Pascal Language system.

To get a listing of files including data files, insert diskette Apple Pascal #1 into drive one and the Pilot diskette containing the files into drive two. Turn the computer on and from the menu of instructions displayed at the top of the screen, choose and type F (for filer). A second set of instructions is displayed so choose and type L (for listing). The message:

```
List directory of:
```

appears on the screen. Type in:

```
#5
```

which is a device number assigned to drive number 2. You can also type in the volume name of the diskette, for instance:

Midisco:

which in this case is the name assigned to the Pilot diskette containing the files that need to be accessed. A complete directory of text and data files will be displayed on the screen.

Using the Pascal system also allows the user to print data files. To do this choose and type T (for transfer) from the filer. The message:

Transfer from?

appears on the screen. Type in the name of the file you want to transfer along with the volume name. For instance:

Mydisk: Skedemo.Data

After a carriage return the message:

Transfer to?

will be displayed. Type in:

Printer:

and your file will be transfered to the printer.

To delete data files choose and type D (for delete) from the filer, followed by the instruction:

Mydisk:skedemo.data

where Mydisk is the volume name and skedemo.data, the file to be deleted. After a carriage return, the question:

Update Directory?

appears on the screen. Type in yes and the file will be deleted. If at the last moment you decide you don't want to delete the file, type No, and your file will remain the same. Text files can be accessed, deleted or printed from the Apple Pilot editor without any inconvenience.

4.3 Text-Files.

Most of the files dealt with in this program are text files. Each text file can occupy only a determined amount of blocks; therefore it is important that the user check on the number of blocks used and remaining before attempting to save the program. Failure to do so will result in text files too long to be saved and the user will get an error message. To avoid this sort of problem the user must divide the file as soon as it is getting to the limit of blocks permitted. To accomplish this, he must save the current file and create a new file with a name such as Part two or the like, as a remainder that it is a continuation of another file. Then he must link both files by means of the L(link) instruction. The Link instruction must be the last command given in the original file and both must be saved in the same diskette unless a multi-drive system is available.

Pilot allows the user to link as many text files or lessons as he wants, and what is more important, the variables created in one can be kept and updated all through the other files. If this is not necessary, the user can always start a new file with completely different variables. To do so, the Link command plus the modifier X must be used. For instance:

LX:Part-two

The modifier X " erases the current file variables and releases all storage space reserved for those variables before starting the new lesson". (Apple. 1980c:77).

A lesson can be linked to a specific label of the other lesson or lessons by using an instruction like:

L:Part-two,Test

where Part-two is the name of the file and Test, the specific label to be linked to. This is particularly useful when having general routines at the beginning of the lesson which need not be accessed right away. This is the case of the Return to Continue routine which most of the times is included at the beginning of each lesson and which is meant to be used from within each lesson. If the original file is not linked to a specific label in the new file, the program will access the Return routine, and since it contains and End instruction execution of the program will stop, mistaking the End instruction of the subroutine for the End of the program.

Since only a pre-determined number of files can be saved in one diskette, it is important to keep track of the space that has been used. If the diskette is already full, the user will get an error message. If this happens it will be necessary to link the current file to a file in another diskette. It is important to point out that the Editor itself occupies a certain space; furthermore the writing error is likely to appear even when all the blocks, according to the message displayed, have not been used. The format of this instruction should be:

```
L:Mydisk:Part-two,Test
```

Where Mydisk is the name of the diskette, Part-two, the name of the file and Test, the specific label to be linked to. Along with the instruction, some of kind of message must be displayed for the user to know that he is suppossed to change diskettes. This is only necessary in one-drive systems.

4.4 Graphics Design.

As was mentioned, graphics can be created in immediate or deferred mode. All the graphics used in this program were created in the editor, which offers a vast amount of possibilities such as drawing circles, squares, filling up areas with determined colors and writing text within the graphics. This last possibility is also available when creating the graphics from the program.

When using the editor, information about coordinates, colors, and number of commands available are displayed at the botton of the screen after typing the letter W. This is particularly useful when there is a need to locate specific positions or to keep track of the commands already used.

Pilot has the capability of letting text appear within the graphics. To do this, type T to get in text mode and insert your text. Pressing the escape key, accepts the instruction and returns to graphic mode. It is important to point out that text typed in immediate mode remains in the graphic and can only be deleted with the E (erasing) instruction, used to erase all the graphic commands. The E instruction only erases the last command accepted, therefore if drawings are not carefully planned before hand, the user might run into problems when trying to erase a part of the graphic drawn at the beginning. It should be pointed out that this is particularly inconvenient and time-consuming when dealing with complex graphics.

If a specific part of the graphic must be changed and there are commands available, a good practice would be to draw a line across the part that needs to be erased, change the color to black and then make a D(draw) instruction. This covers the spot with black which for all purposes is the same as erasing.

4.4.1 Text-Windows.

Since normally any graphic should have a text window for displaying text while running the program, it is advisable that the user determine the coordinates to set it before hand. To do this, simply move the cursor to the vertical or horizontal line where the text window should be, type W and from the menu displayed, take note of the X and Y coordinates to use them later in the program when recalling the graphic and actually setting the text window.

4.4.2 Quick and Slow Draws.

There are two ways to display graphics in Pilot, the quick draw, which automatically recalls and displays the entire picture or the slow draw, which recreates the picture on the screen step by step. The slow picture type is very convenient when the graphics designed are not very complex. If they are, a quick draw would be more effective, since having a graphic recreated on the scree step-by-step is very time-consuming. Nonetheless, there is a disadvantage to quick draws, namely that they occupy an enormous amount of disk space, i.e. approximately 15 to 20 blocks for each figure.

4.4.3 Recalling Graphics.

When recalling a graphic from the program, an instruction like the following must be executed:

```
GX:Mydisk:Skeleton!
```

where Mydisk is the name of the diskette and skeleton the name of the file. The exclamation mark is used to determine whether or not it is a quick draw. This has to be specified when first creating the graphic. The name of the diskette is only necessary when the graphics are stored in a diskette different from where the main program is. If the graphics and the main program are stored in the same diskette, the instruction should be:

```
GX:Skeleton!
```

In order to set the text window when recalling the graphics, use the instruction:

```
G:V0,39,21,23
```

In this case, the first two values 0 and 39 set the horizontal position while 21 and 23 set the vertical position. In this example the result is a text window at the bottom of the screen, 39 columns wide and 3 rows high.

As was mentioned above,the text window can be placed at any desired position in the screen. Its location should be planned beforehand, perhaps even before planning the graphics, so that the teacher can be certain that there will be enough space to display text in. Not doing so results in text windows to small to display more than two or three lines at a time, which is rather inconvenient.

If a Return to Continue routine should be used while displaying text in the text window, the teacher must make sure that a G:ES rather than a G:ES0 instruction is included in the routine. G:ESO causes the whole screen to be erased including the graphics while G:ES, only erases whatever text is displayed in the screen at the moment. This does not include text that has been typed in while creating the graphics in the editor.

5. PROJECT EVALUATION

Several conclusions were made regarding this project. First, the Apple Pilot Language itself is easy to learn therefore, even without computer background a teacher can easily develop simple computer assisted instruction programs. Nevertheless, documentation to develop more complex programs is not sufficient. This would require that the teacher be more familiar with computer programming techniques.

The fact that the system uses floppy diskettes and loads only a determined amount of information into the memory of the computer at a time, results in a considerable slow down when running the program. This is also due to the fact that the program itself is long and rather complex since it has to check in a vast amount of conditions before actually accessing the files where the objects of the instructions are saved.

In terms of logic, most of the program could have been saved under one single file name, but unfortunately Pilot only allows an amount of eigtheen blocks for each file. Under these circumstances, the program had to be divided every time the space permitted was used. Nonetheless, it should be pointed out that those divisions were logically made in the sense that they correspond to routines that in one way or another could be considered independent. This is particularly useful if the user wants to implement new material since he would only have to deal with specific files rather than with the whole program.

Pilot offers a high capability for graphic design and use. Unfortunately complex graphics occupy and immense amount of disk space which narrows the space availability for text programming. This results in the need to use several diskettes, which for the purpose of programming and use is rather inconvenient.

Contrary to other systems, Pilot has a built in command to print out text files. This is a disadvantage in the sense that only entire files can be printed even if only a couple of pages are needed. Although it is possible to print out graphics, it is very inconvenient since special equipment must be used, specifically a Sylent type printer which, needless to say, is not always available.

In general the system is easy to use and its programms run nicely. Implementing the system with a hard disk would be the perfect solution for speeding up the running. The fact that the Apple Pascal System can be used to access the Pilot files is a considerable advantage since the Pascal system comes with an editor which is much more effective than the one in Pilot.

Appendices

REFERENCES AND BIBLIOGRAPHY

Apple Computer Inc. 1980(A). Apple Pascal Operating System Reference Manual.

Apple Computer Inc. 1980(B). Apple Pilot Editors Manual.

Apple Computer Inc. 1980(C). Apple Pilot Language Reference Manual.

Apple Computer Inc. 1980(D). Pilot Animation Tools.

Lado, Robert and Fries C, Charles. 1960. English Sentence Patterns. Understanding and producing English grammatical structures. An oral approach. Ann Arbor. The Univerity of Michigan Press.

Poirot L, James. 1980. Computers and Education. Sterling Swift Publishing Company.

I. Program Listings

I-A Wordquest

```
r:WordQuest
r:Programmed by Luz Zahydee G. Minnick
r:October 1982


c:z=0
c:e=0
c:l=10
c:g=50
c:q=15
c:f=0
c:c=0
c:w=0

r:Initializing review array
d:r(4)
c:r(1)=0
c:r(2)=0
c:r(3)=0
c:r(4)=0


r:Initializing question counter array
d:m(4)
c:m(1)=0
c:m(2)=0
c:m(3)=0
c:m(4)=0

r:Initializing array label questions
:answered
d:p(4)
c:p(1)=0
c:p(2)=0
c:p(3)=0
c:p(4)=0

r:Initializing hints array
d:h(4)
c:h(1)=0
c:h(2)=0
c:h(3)=0
c:h(4)=0


r:initializing obstacle label array.
d:o(2)
c:o(1)=0
c:o(2)=0

j:start

*Return
g:m0,3
```

```
th:...... Press return to continue
:.........
a:
g:es0
g:v
e:

*start
d:n$(30)

gx:graph1:entrada!
g:v0,39,21,23
t:.. W E L C O M E    S T R A N G E R ...


s:20,25;24,25;27,100;20,25;24,25;27,100
:;20,25;22,25;24,25;27,25;20,25;22,25;2
:4,25;27,25;24,25;27,25;32,100

t:
u:return
t:
t:

t:Excuse me, are you a male or a
t:
th:female?
pr:u
a:
m:FEMALE
jy:lady
jn:man

*lady
t:
t:OH!...an adventurous lady......what
t:
t:a wonderful surprise!. Brave ladies
t:
t:are not seen very often around these
t:
t:remote lands.
t:
t:
t:
t:By what name should I call you, fair
t:
th:maiden?
a:$n$
 t:
t:

u:return
```

```
j:talk

*man
t:
t:
t:AHA! A brave gentleman searching
t:
t:for adventure and treasure on these
t:
t:strange lands.....................
t:
t:By what name should I call you, brave
t:
th:adventurer?
a:$n$
p:
t:
t:
u:return
j:talk

*talk
t:
th:Have you run this lesson before?
pr:u
a:
m:YES!YEA!SURE!
jy:luck
u:return

t:
t:Then, let me introduce myself,
t:
t:I am the Wizard of WordQuest.
t:
t:Since you haven't been around here
t:
t:before, I assume you will need some
t:
t:information in order to find your
t:
t:way inside the dungeons.
u:return


t:In your trip through WordQuest, you
t:
t:will find rooms, secret doors and
t:
t:corridors, cliffs, treasure,and
t:
t:hints.
t:
```

```
t:You will also meet a diversity of
t:
t:monsters and fabulous beasts who will
t:
t:not let you out unless you answer
t:
t:their questions.
u:return
t:$n$
t:
t:Your goal is to find your way out
t:
t:**** ALIVE ****
t:
t:     ***** ALIVE ****
t:
t:           ***** ALIVE ****
t:
t:                            and
t:
t:    with as much GOLD as possible
t:
t:    *****************************
u:return
t:
t:
t:A map of the caves will be dis-
t:
t:played along with a pair of inter-
t:
t:secting cross-hairs.
t:
t:By using the game controls
t:
t:you can move the intersection
t:
t:of the cross-hairs to any of the
t:
t:labeled and or unlabeled asterisks
t:
t:marked in the map.  Game control 0
t:
t:sets the horizontal position and
t:
t:game control 1, the vertical posi-
t:
t:tion.
t:
u:return
t:
t:When the corss-hairs are centered on
t:
t:the desired location, press the but-
```

```
t:
t:ton on either game control, then:
t:
t:   ********************************
t:
t:*** SURPRISING THINGS WILL HAPPEN ***
t:
t:   ********************************
t:
t:
t:
u:return
t:You can check on the amount of gold,
t:
t:life points, questions left, and
t:
t:your score by moving the cross-
t:
t:hairs to the asterisk labeled
t:
t:INFORMATION. BOOK displays a
t:
t:general review of the subject and
t:
t:QUIT allows you to stop and leave
t:
t:the program.
t:
t:
t:
u:return
t:
t:
t:
t:
t:You are about to become the HERO or
t:
t:the VICTIM  of one of the strangest
t:
t:adventures that has ever befallen a
t:
t:mortal.
t:
t:
t:
u:return

*luck
g:es0
t:
t:$n$
t:
t:
```

```
t: * G *
t:
t:   * O *
t:
t:     * O *
t:
t:       * D *
t:
t:
t:            * L *
t:
t:              * U *
t:
t:                * C *
t:
t:                  * K *
t:
t:
u:return
t:
t:
t:
t:
t:
t:   YOU ARE BEING TELEPORTED TO
t:
t:
t:       LEVEL 1 OF WORDQUEST
t:
t:
t:
t:
l:map1,map
e:
```

I-B Mapl

```
*return

t:****** Press return to continue
:******

w:2
a:

g:es
e:

r:subroutine to display information

*inform
g:v
g:es0
t:********************************
t:
t:       General Information
t:
t:
t:
t:    Gold             = #g
t:
t:    Life             = #l
t:
t:    Questions left   = #q
t:
t:    Right answers    = #c
t:
t:    Wrong answers    = #w
t:
t:
t:*********************************
u:return
g:es0
j(e=1):print
j(e=0):map
*print
t:
t:
t:
t:
t:
t:
t:
t: ** Be patient $n$ ,
t:
t:    Mrs. Bones is checking you out
t:
t:         .........................
l:skedemo
e:
```

```
*map
g:es0
gx:map1!
g:v0,39,21,23

j(p(1)=0):label2
g:m44,237
g:t? set #1

*label2
j(p(2)=0):label3
g:m286,344
g:t? set #2


*label3
j(p(3)=0):label4
g:m404,412
g:t? set #3

*label4
j(p(4)=0):label5
g:m48,485
g:t? set #4



*label5
j(o(2)=0):label6
g:m48,379
g:tRuins

*label6
j(o(1)=0):label7
g:m246,260
g:tcliff

*label7
j(q>0):try
g:m46,96
g:tE X I T

*try


t:Move the paddles where you want to
t:go $n$ ,  and press the button.

w:3
g:es

ap:
```

```
u:return
t:By reading those pages you'll have a
:general idea of the material covered
:by the reviews and included in the
:questions.
u:return
t:
t:If you are not sufficiently prepared
:I suggest that you go through that
:material right now, otherwise you
:might loose your life sooner than you
:expect.

t:Do you want to read it now?
t:
t:
th:?
pr:u
a:
m:YEA!YES!ALL RIGHT!O.K!
ly:sumup,start
jn:map1,map


*exit
j(q=0):out
j(q<>0):back

*back
g:es0
g:v

t:
t:This is the passage to the outside of
t:
t:the cave but it will be locked up
t:
t:until you answer all the questions.
t:
t:When the right time comes, the word
t:
t:EXIT will be displayed and you will
t:
t:be free to go.
t:
u:return
j:map

*out
g:es0
g:v
t:
t:
```

```
t:
t:**         Congratulations         **
t:
t:**           $n$                   **
t:
t:**   You made it outside the cave  **
t:
t:           Please type in
t:
t:           THE MAGIC WORD
t:
th:          ?
pr:u
a:
m:ABRACADABRA
jn:chance
jy:afuera
*chance
g:es0
t:
t:Don't try to fool me charlatan, that
t:
t: is not the magic word........
t:
t:
t:Nevertheless I think  I'll let you
t:
t:out. After all you've already
t:
t:answered all the questions.
t:
u:return

*afuera

gx:exit!
u:return
c:e=1
j:inform


*quit
g:es0
g:v
t:$n$ ,
t:
t:Too bad you don't have the time to
t:
t:go through the whole adventure. Any-
t:
t:way, before we set you free we want
t:
t:you to see how well you did.
```

```
t:
u:return
gx:exit!
u:return
c:e=1
j:inform



*cliff
gx:graph1:cliff!
g:v0,39,0,5
c:o(1)=1
t:
t:        *******  DANGER ********
u:return
*busca
ap:
c:h=%x
c:v=%y
j(h>434 and h<442 and v>4 and v<12):map
j(h>202 and h<210 and v >359 and
:v<367):cliff1
j(h>350 and h<358 and v>215 and
:v<223):cliff2
j(h>542 and h<550 and v >153 and
:v<161):cliff3
t:That was not close enough $n$ , try
:again.
w:6
g:es
j:busca

*cliff1
t:$n$ , I told you it was dangerous,
:you have just fallen down the cliff.
t:You have lost 5 life points......
c:l=l-5
u:return
j:busca

*cliff2
t:I think you've found something
:valuable. How about 5 pieces of gold
:to get it?
pr:u
th:?
a:
m:YES!YEA!ALL RIGHT!O.K
jy:page3
tn:I hope you won't regret it later
u:return
jn:busca
```

```
*page3
t:$n$ , I think that was a very good
:deal.  Now you have another hint in
:your magic book.Do you want to read
:it immediately
pr:u
th:?
a:
m:YES!YEA!ALL RIGHT!O.K
c:r(3)=1
c:g=g-5
c:h=0
ly:review,three
jn:cliff

*cliff3
c:r(4)=1
c:h=0
t:Today must be your lucky day $n$ ,
:You've just found a chest containing
t:10 pieces of gold and an old
:manuscript where a hint for question
:#3 is written.
c:g:g+10
u:return
t:do you want me to decipher the
:manuscript right now?
th:?
pr:u
a:
m:YES!YEA!ALL RIGHT!O.K.
ly:review,four
jn:busca


*ruins

gx:graph1:stairs!
gx:graph1:stairscomp
g:v0,39,19,23
c:o(2)=1
t:
t:*** OMENS OF DOOM *** WATCH OUT ***

*search
ap:
c:h=%x
c:v=%y
j(h>378 and h <386 and v>455 and
:v<463):map
j(h>0 and h<8 and v >491 and
:v<500):hint1
```

```
j(h>226 and h<234 and v>231 and
:v<239):hint2
j(h>528 and h<536 and v >383 and
:v<391):hint3
t:That was not close enough $n$ , try
:again.
w:6
g:es
j:search



*hint1
g:es
pr:u
th:$n$ , You have found something, Are
:you willing to pay 5 pieces of gold to
:see what it is?
a:
m:YES!YEA!ALL RIGHT!O.K.
jy:page1..
tn:You'll never know what you missed...
u:return
jn:search

*page1
g:es
t:You have found page1 for your magic
:book....You never know when you'll
:need it.........
u:return
c:r(1)=1
c:g=g-5
c:h=0
t:Do you want to read it right away?
t:
th:?
pr:u
a:
m:YES!YEA!SURE!RIGHT!OK!O.K
ly:review,one
jn:ruins

*hint3
c:r(2)=1
c:g=g+10
c:h=0


t:good for you $n$ , you have found a
:precious stone worth 10 gold pieces
:plus another hint. Do you want to see
:what this one is about?
```

```
pr:u
th:?
a:
m:YES!YEA!ALL RIGHT!SURE!O.K
tn:I hope you won't regret it
jn:search
ly:review,two


*hint2
t:$n$ , I am afraid you've just
:drowned. You got to be more careful.
t:One can not be so ambitious......
c:l=l-1
u:return
j:search
e:
```

I-C Monstersl

```
*return
g:m0,2

g:t************* Press return to continue

a:

g:es
g:m0,2
g:t.....................................
:..

e:


*check1
t:$n$ ,
t:You don't have the hints  for these
t:questions. If you want, you can go
:ahead and answer the questions but you
:can also go back to the map and look
:for more hints.
th:Would you rather go back?
pr:u
a:
m:YES!YEA!O.K!OK!
ly:map1,map
g:es0
g:v
ln:question,set1

*check2

t:$n$ ,
t:You might remember that you've
:already found the hints for question
:set #1.
t:Do you want to read them now?
pr:u
th:?
a:
m:YES!YEA!SURE!RIGHT!OK!O.K

g:es0
g:v
c:h=1
ly:review,one
ln:question,set1



*chekd1
```

```
t:$n$ , you haven't found the hints for
:question set #2.
u:return
t:If you want to win DRAGONA'S bet, I
:recommend that  you go back and  look
:for more hints. It is up to you.
th:Are you going back?
pr:u
a:
m:YES!YEA!RIGHT!SURE!OK!O.K
t:
ly:map1,map
g:es0
g:v
ln:question,set2


*chekd2
t:Fortunately, you already have the
:hints for this question set.
t:Do you want to read them now
th:?
pr:u
a:
m:YES!YEA!OK!O.K!ALL RIGHT!RIGHT
g:es0
g:v
c:h=1
ly:review,two
ln:question,set2


*chekh1
t:I was just looking through your magic
:book and I discovered that the hints
:for this question set aren't in there.

u:return
t:
th:You are lucky that Cerberus will
:give you the choice to back up. Do you
:take it?
pr:u
a:
m:YES!YEA!SURE!OK!O.K!ALL RIGHT
g:es0
g:v
ly:map1,map
ln:question,set3

*chekh2

th:With such a smart competitor, you
```

```
:must be glad you have the necessary
:hints. Are you going to read them?
pr:u
a:
m:YES!YEA!OK!O.K!
g:es0
g:v
c:h=1
ly:review,three
ln:question,set3

*chekb1
t:Unfortunately your search through the
:caves hasn't been very fruitful, at
:least until now....
t:
t:But don't dispair, the magician will
:let you go back and return when you
:are better prepared.
u:return
t:
th:Do you want to go
t:back?
pr:u
a:
m:YES!YEA!ALL RIGHT!OK!O.K
g:es0
g:v
ly:map1,map
ln:question,set4

*chekb2
t:How about reading the hints for
t:question set #4? I know you have them
:in your magic book.
t:
th:Do you want to?
pr:u
a:
m:YES!YEA!ALL RIGHT!OK!O.K
g:es0
g:v
c:h=1
ly:review,four
ln:question,set4

*eskele
c:p(1)=1
gx:graph1:skeleton!
g:v0,10,0,23
j(m(1)=1):sk2
t:
t:"I AM MR. BONES. FOR DISTURBING MY
```

```
:WEARY BONES YOU WILL HAVE TO ANSWER MY
:QUESTION, OR ELSE I WILL TAKE YOUR
:LIFE.
t:(MAYBE IT'S WORTHWILE, MAYBE IT
:ISN'T.)
u:return
j(r(1)=0):check1
j(r(1)=1):check2


*sk2
t:$n$ , you have already answered this
:set of questions. You must go back.
u:return
l:map1,map

*score1
gx:graph1:skeleton!
g:v0,10,0,23
j(c1=4):win1
t:$n$ , you got #c1  right answers in
:this set. So far you have #l  life
:points.
u:return
l:map1,map

*win1
t:"I SEE YOU GOT ALL OF THEM RIGHT
:ON THE FIRST TRY. YOU MUST BE
:SMART.....
t:THANKS TO YOU MY WEARY OLD BONES WILL
 t:FINALLY TURN BLACK..............


t:
u:return
t:
t:
t:
t:
t:"FOR DUST I AM AND UNTO DUST I SHALL
:BECOME"..............
t:
t:!AY......AY.......AY.....AY...!!!!!!
g:es8
t:
t:So far you have #l  life points.
t:
u:return
l:map1,map

*dragon
c:p(2)=1
```

```
gx:graph1:dragon!
g:v12,39,17,23
j(m(2)=1):dra2
t:I AM DRAGONA THE BEAUTIFUL DRAGON.........
w:2
t:I BET YOU MY LOVELY LOOKS THAT YOU
:WON'T BE ABLE TO ANSWER MY QUESTIONS....
u:return
t:IF YOU DO, I WILL BECOME UGLY, BUT IF
:YOU DON'T I'LL BE MORE LOVELY THAN
:EVER..............
u:return
j(r(2)=0):chekd1
j(r(2)=1):chekd2

*dra2
t:$n$ ,I AM LOVELY AND BEAUTIFUL BUT
:NOT KIND ENOUGH TO LET YOU ANSWER THIS
:SET OF QUESTIONS AGAIN.GO BACK......
u:return
*score2
gx:graph1:dragon!
g:v12,39,17,23
j(c1=3):win2
t:YOU GOT #c1  RIGHT ANSWERS. THAT
:WASN'T GOOD ENOUGH TO BEAT ME. THANKS
:TO YOU I'LL BE BEAUTIFUL FOR EVER....
u:return
l:map1,map

*win2
t:I never thought you were so smart,
:seems like I will have to pay for my
:vanity..........
u:return
g:es8
t:!OH NO.!
t:NOW I'VE BECOME DRAGONA THE UGLIEST
w:3
t:!AY....AY.....AY.......
w:3
t:You have gained #1  life points so
:far.
u:return

l:map1,map

*heads
c:p(3)=1
gx:graph1:cerberus!
g:v0,39,20,23
j(m(3)=1):head2
t:AH HA!  I see I have have a visitor.
```

```
t:
u:return
t:My name is CERBERUS AND I HAVE THE
t:BRAINS OF THREE, SO WATCH OUT....
u:return

j(r(3)=0):chekh1
j(r(3)=1):chekh2

*head2
t:$n$ , you again? you are not allowed
:to come here for a second time. Go
:back where you belong......
w:8
l:map1,map



*score3
gx:graph1:cerberus!
g:v0,39,20,23
j(c1=4):win3
t:$n$ , you got #c1  correct answers in
:this set. So far you have #l  life
:points.
w:5
l:map1,map

*win3
t:Well, I will have to accept that YOU
:ARE GOOD. You got #c1  out of 4 in
:this set.
u:return

w:8
g:es8
t:ay...ay.....ay
t:You have gained #l  life points
u:return
l:map1,map

*brujo
c:p(4)=1
gx:graph1:brujo!
g:v0,10,0,23
j(m(4)=1):bru2
t:I am the MAGICIAN and I have a little
:gift for you,but you can only have it
:after you answer my questions.
u:return
j(r(4)=0):chekb1
j(r(4)=1):chekb2
```

```
*bru2
t:$n$ , walk back where you came from.
:You are not allowed to answer these
:questions again.
u:return
l:map1,map

*score4
gx:graph1:brujo!
g:v0,10,0,23
j(c1=4):win4
t:$n$ , you got #c1  correct answers.
:Since I honor my promises, here is the
:little gift I was talking about.
t:
u:return
t:Write this down some place, for it is
:the magic word which will let you out
:of the cave.
t:
t:the word is:
t:
t:ABRA-
t:CADABRA
u:return
l:map1,map

*win4
t:I see you are a wise person,
:otherwise you wouldn't have gotten all
:the answers correct. This is your
:reward, the magic word that will let
:you out of the cave.
:
t:
t:ABRACADABRA
u:return
t:
t:Now I shall go where I belong.....
t:
t:You have gained #l life points so
:far.
g:es8
u:return

l:map1,map
```

I-D Sumup

```
*return
g:m0,0
t:**** Press return to continue ****
a:
g:es0
e:

*start
g:es0

g:v
t:$n$ ,
t:
t:
t:The material you are about to cover
t:
t:deals with COMPARATIVES. It is a
t:
t:review of the different degrees of
t:
t:comparisons for adjectives and
t:
t:adverbs. Let's consider the following
t:
t:examples:
t:
t:
t:
t:   John is older than Mary.
t:   Mary is young
t:   Paul is the oldest of the group
t:
t:
u:return
t:
t:based on this we can say that there
t:
t:are different degrees of comparison
t:
t:for adjectives and adverbs, meaning
t:
t:that an object (animate or inanimate)
t:
t:can have more, as much as, less or
t:
t:the most of a quality in relation to
t:
t:other objects with which it is
t:
t:compared.
t:
u:return

t:Most adjectives and adverbs have
```

```
t:
t:three degrees of comparison:
t:
t:   the positive degree
t:   the comparative degree
t:   the superlative degree.
t:
t:The positive degree expresses simply
t:
t:the quality of an object without re-
t:
t:ference to any other object. for
t:
t:instance:
t:
t:     John is intelligent
t:     This car is expensive
t:
t:Here we are just mentioning a quality
t:
t:about John and about a car.
u:return
t:
t:The comparative degree shows that an
t:
t:object has more or less of a quality
t:
t:than some other object with which it
t:
t:is compared. For instance:
t:
t:     Peter is richer than John
t:  My car is more expensive than yours
t:
t:
t:Here we are actually comparing how
t:
t:rich Peter is in relation to John or
t:
t:how expensive my car is in relation
t:
t:to yours.
t:
u:return
t:
t:The superlative degree expresses the
t:
t:greatest or least amount of a quality
t:
t:that is found among three or more
t:
t:objects compared.  For instance:
t:
```

```
t: John is the smartest in the family.
t:
t:The comparative is used for a compa-
t:
t:rison between two, while the superla-
t:
t:tive is required where more than two
t:
t:are involved.
t:
u:return
t:
t:The superlative is sometimes used for
t:
t:a comparison between two, as in:
t:
t:   He is the youngest (of the two
t:   brothers),
t:
t:but this use is considered loose and
t:
t:informal.
t:
u:return
t:
t:Comparisons can also express lesser
t:
t:rather than greater degrees of a
t:
t:quality:
t:
t: John is less intelligent than Paul.
t:
t:  His grades are the lowest in the
t:  class.
t:

t:Let's see if you are following me so
t:
t:far. Write in front of each of the
t:
t:following sentences whether it should
t:
t:be classified as positive, comparati-
t:
t:ve or superlative:
t:
u:return
t:Type positive, comparative or superla-
t:tive:
*uno
t:
t:I am older than my sister.
```

```
th: ?
pr:u
a:
m:COMPARATIVE
ty:That is fine.
jy:dos
t:That is not correct. Try again
t2:comparative is the correct answer.
j2:dos
j:uno
t:
u:return
*dos
t:
t:That picture is beautiful.
th:?
pr:u
a:
m:POSITIVE
ty:right
jy:tres
t1:wrong. Try again.
t2:positive, is the correct answer. We
t2:are just stating a quality of the
t2:picture without comparing it to any
t2:other picture.
j2:tres
j:dos
u:return
t:
*tres
t:
t:This is the most interesting book  I
t:have ever read.
th:?
pr:u
a:
m:SUPERLATIVE
ty:right
jy:nota
t1:wrong. Try again.
t2:The correct answer is superlative.
j2:nota
j:tres


u:return
*nota
t:
t:$n$ , as you wander through the
t:
t:dungeons of WordQuest, you will find
t:
```

```
t:more hints to  prepare you better to
t:
t:fight the monsters.
l:map1,map



e:
```

I-E Review

```
 *return
g:m0,0
t:**** Press return to continue ****
a:
g:es0
e:



*one
g:es0
g:v
t:  This is a hint for question set #1.
t:
t:Adjectives and adverbs of one
t:
t:syllable usually add the inflectional
t:
t:endings ER to form the COMPARARIVE
t:
t:and EST to form the SUPERLATIVE:
t:
t:  tall        tallER    tallEST
t:
t:  loud        loudER    loudEST
t:
t:  For example:
t:
t:  Mary is the tallEST girl in the
t:  family.
t:
t:  Girls are usually loudER than boys.
t:
t:
u:return
*aa
g:es0
t:
t:Complete the following sentence with
t:
t:the appropriate  form:
t:
t:Children are ---- (loud) than adults.
t:
th:?
pr:u
a:
m:LOUDER
t:
ty:You are doing fine.
jy:ab
t:
t1:Try again, $n$  . Remember, it is a
:monosyllabic adjective.
```

```
t:
t2:No, the right answer is LOUDER. A
:monosyllabic adjective is used to
:compare two course subjects.
j2:ab
j:aa

*ab
t:
t: Calculus is -- ----- (hard) class
t: in the high school curriculum.
t:
th:?
pr:u
a:
m:THE HARDEST
t:
ty:You are doing fine.
jy:ac
t:
t1:Not quite right, try again.
t:
t2:The correct answer is THE HARDEST.
j2:ac
j:ab
*ac
u:return
t:
t:Words of three or more syllables are
t:
t:prefaced by the words MORE for the
t:
t:COMPARATIVE degree and EST for the
t:
t:SUPERLATIVE degree:
t:
t:          beautiful
t:
t:     more beautiful
t:
t:     most beautiful
t:
t:For instance:
t:
t:Learning Japanese  is MORE DIFFICULT
t:than learning English.
t:
t:This is THE MOST RELIABLE machine you
t:can buy in the market.
t:
u:return
*ad
t:
```

```
t:Complete the following sentence with
t:
t:the appropriate comparative or super-
t:
t:lative form:
t:
t:  Roses are --- --- ------ (beautiful)
t:  flowers.
th:?
pr:u
a:
m:THE MOST BEAUTIFUL
t:
ty:You are doing fine.
jy:ae
t:
t1:That is not right, try again
t:
t2:THE MOST BEAUTIFUL is the answer.
j2:ae
j:ad

*ae
t:
t:Reading books is --- --- (exciting)
t:than watching television.
th:?
pr:u
a:
m:MORE EXCITING
t:
ty:Very well.
jy:af
t:
t1:I'll give you another chance.
t:
t2:MORE EXCITING is the correct answer.
j2:af
j:ae
*af
l(h=0):map1,ruins
l(h=1):question,set1

*two
g:es0
g:v
t: This is a hint for question set #2.
t:
t:With two syllable modifiers, the
t:
t:choice  depends  on the rhythm
t:
t:and harmony of sound.
```

```
t:
t:   happy       happier    happiest
t:
t:   eager   more eager   most eager
t:
t:  His children are politER
t:
t:  His children are MORE polite.
t:
u:return
t:
t:Common disyllabic adjectives that can
t:
t:form the comparative with ER are
t:
t:those ending in an unstressed vowel.
t:
t:-y:funny, noisy, wealthy, friendly.
t:
t:-ow: hollow, narrow, shallow.
t:
t:-le: gentle, feeble, noble
t:
t:-er,-ure: clever, mature, obscure.
t:
t:Common adjectives outside these four
t:
t:categories that can take inflectional
t:
t:forms include:
t:
t:  common, handsome, polite, quiet
t:  wicked.
u:return
*ag
t:
t:Complete the following sentences:
t:
t: My jokes are ----- (funny) than
t: yours.
th:?
pr:u
a:
m:FUNNIER!MORE FUNNY
t:
ty:That is right.
jy:ah
t:
t1:Try again $n$ .
t:
t2:Since FUNNY is a two syllable
:adjective it can take both forms.
```

```
:Nevertheless, you must remember that
:MORE FUNNY is not very commonly used.

j2:ah
j:ag

*ah
t:
t:John is ------ (quiet) than a mouse.
th:?
pr:u
a:
m:QUIETER!MORE QUIET
t:
ty:You are doing very well.
jy:hh
t1:Try again $n$
t:
t2:The right answer is either QUIETER
:or MORE QUIET since it is a disyl-
:labic adjective.
j2:hh
j:ah
*hh
l(h=1):question,set2
l(h=0):map1,ruins

*three
g:es0
g:v
t: This is a hint for question set #3
t:
t:Some modifiers have irregular forms
t:
t:of comparison:
t:
t:POSITIVE   COMPARATIVE  SUPERLATIVE
t:
t:bad        worse        worst
t:
t:far        farther      farthest
t:           further      furthest
t:good,well  better       best
t:
t:little     littler      littlest
t:
t:much,many  more         most
t:
t:old     older,elder   oldest,eldest
t:
t:
t:The choice between FARTHER and
t:
```

```
t:FURTHER depends mainly on the
t:
t:context.
t:
u:return
*ai
t:
t:Complete the following sentences:
t:
t:The movie was a lot ----- (good)
t:than the book.
t:?
pr:u
a:
m:BETTER
ty:Very well
jy:aj
t1:How about another try.
t:
t2:BETTER is the correct answer.
j2:aj
j:ai
*aj
t:
t:This is the ------ (bad) paper you
t:have ever written.
th:?
pr:u
a:
m:WORST
ty:That is right.
jy:ak
t1:Remember it is an irregular form.
t2:The right answer is WORST.
j2:ak
j:aj
*ak
u:return
l(h=1):question,set3
l(h=0):map1,cliff

*four
g:es0
g:v
t: This is a hint for question set #4
t:
t:It is important to consider compa-
t:
t:risons based not only on lack of
t:
t:equivalence, such as the expressed
t:
t:by MORE and LESS, but comparisons
```

```
t:
t:based on other relationships, notably
t:
t:those of equivalence as expressed by
t:
t:AS..AS, and LIKE.
t:
u:return
t:
t:Use LIKE, and AS....AS to compare two
t:
t:persons or things. For instance:
t:
t: Mary dresses AS elegantly AS Jane.
t:
t: He works LIKE a horse.
*al
t:
t:Combine the following two sentences
t:
t:into one by using AS .. AS:
t:
t: Mary speaks clearly.
t: John speaks clearly.
th:?
pr:u
a:
m:MARY SPEAKS AS CLEARLY AS JOHN
ty:That is very good.
jy:am
t1:Give it another try.
t:
t2:the correct form would be MARY
:SPEAKS AS CLEARLY AS JOHN.
j2:am
j:al
*am
t:
t:Type true or false depending on
t:
t:whether the next sentence is correct
t:
t:or not:
t:
t:John behaves just LIKE his father.
t:
th:
pr:u
a:
m:TRUE!T
ty:Very good.
jy:an
t1:Try again $n$ ,I am sure you'll get
```

```
:it right then.
t2:The sentence is correct, therefore
:the answer is TRUE.
j2:an
j:am
*an
u:return
t:
t:In terms of logic, a person or thing
t:
t:should not be compared with a part or
t:
t:a quality of another person or thing:
t:
t:
t:  The GROWL of a tiger is more
t:
t:  frightening than an EAGLE.
t:
u:return
t:
t:This sentence compares a sound with a
t:
t:bird. A consistent and logical
t:
t:sentence would compare sound with
t:
t:sound or bird with bird:
t:
t: The growl of a tiger is more
t:
t: frightening than the scream of an
t:
t: eagle.
t:
u:return
*ao
t:
t:Type true or false depending on
t:
t:whether the next is a logical
t:
t:sentence:
t:
t:  The song of a bird is more
t:  harmonious than a parrot.
t:
th:
pr:u
a:
m:FALSE!F
ty:That is right.
jy:ap
```

```
t1:Hou about another try.
t:
t2:The sentence is incorrect, therefore
:the answer was FALSE.
j2:ap
j:ao
*ap
l(h=1):question,set4
l(h=0):map1,cliff
e:
```

I-F Question

```
*return
g:m0,4
g:t..... Press return to continue
:.......
a:
g:es0
e:
r:right answer routine
*right
r:right answer sound
s:20,50;29,50;20,50;29,50;20,50;29,50;2
:0,50;29,50
t:
t:
t:
t:********  C O R R E C T **********
u:return
c:c=c+1
c:c1=c1+1
c:l=l+5
c:q=q-1
e:

r:wrong answer routine
*wrong
r:wrong answer sound
s:8,255;1,255
t:
t:
t:
t:   ******* I N C O R R E C T *******

c:w=w+1
c:l=l-5
c:q=q-1
e:

*set1
c:m(1)=m(1)+1
c:c1=0
t:......This is question set #1......
t:
*one
t:
t:Complete the following sentence:
t:
t:Paul works ----- (hard) than his
t:brother.
t:
th:      ?
pr:u
a:
m:HARDER
uy:right
```

```
jy:two
u:wrong
t:
t:
t:
t:The correct answer is HARDER .
:Adverbs of one syllable add the
:iending ER to form the comparative.

u:return

*two
t:
t:Complete the following sentence:
t:
t:He is the ---- (old) in the family.
t:
th:      ?
pr:u
a:
m:OLDEST
uy:right
jy:three
u:wrong
t:
t:
t:The correct answer is OLDEST. One
:syllable adjectives add the ending
:EST to form the superlative.

u:return

*three
t:
t:Complete the following sentence:
t:
t:Jane is --- --- (interesting) than
: her sister.
t:
th:      ?
pr:u
a:
m:MORE INTERESTING!LESS INTERESTING
uy:right
jy:four
u:wrong
t:
t:
t:The correct answer is either MORE
:INTERESTING  or LESS INTERESTING.
t:Polysyllabic adjectives use MORE THAN
:or LESS THAN to form comparisons.
u:return
```

```
*fcur
t:
t:Complete the following sentence:
t:
t:The small ones are the --- (cheap)
:and ----- ----- (beautiful) cars.
t:
th:      ?
pr:u
a:
m:CHEAPEST,MOST BEAUTIFUL
uy:right
ly:monsters1,score1
u:wrong
t:
t:
t:The correct answers are CHEAPEST and
t:MOST BEAUTIFUL. Adjectives of one
:syllable form the superlative by
:adding EST while adjectives of more
:than two syllables use MOST.
u:return
l:monsters1,score1

*set2
c:m(2)=m(2)+1
c:c1=0
t:....This is question set #2 .....

*five
t:
t:
t:Which of the following sentences is
:grammatically incorrect:
t:
t:1.This was the happiest birthday I
:   have ever had.
t:
t:2.Mary is friendlier than Betty.
t:
t:3.The other car is costlier than this
:   one.
t:
t:4.Mr. Brown walks more slowlier than
:   Mr. Smith.
t:
th:      ?
pr:u
a:
m:FOUR!4
uy:right
jy:six
```

```
u:wrong
t:
t:
t:Sentence #4 is grammatically
:incorrect since it uses both, MORE and
:the ending ER to form the comparison.
u:return

*six
t:
t:
t:True or False.
t:
t:The comparative form of POLITE can be
:either POLITER or MORE POLITE.
t:
th:      ?
pr:u
a:
m:TRUE
uy:right
jy:seven
u:wrong
t:
t:
t:The answer is TRUE. Since POLITE is a
:two syllable adjective it can take
:both forms.
u:return

*seven
t:True or False:
t:
t:The superlative form of DIFFICULT is
:either THE MOST DIFFICULT or THE DIF-
t:FICULTEST.
t:
th:      ?
pr:u
a:
m:FALSE
uy:right
ly:monsters1,score2

u:wrong
t:
t:
t:The answer is:FALSE. The option to
:use the ending EST or THE MOST to form
:the superlative applies only to
:disyllabic adjectives or adverbs.

u:return
```

```
l:monsters1,score2

*set3
c:m(3)=m(3)+1
c:c1=0
*nine
t:  This is question set #3

t:
t:Complete the following sentence:
t:
t:She speaks ----- (bad) than she talks.
t:
th:    ?
pr:u
a:
m:WORSE
uy:right
jy:ten
u:wrong
t:
t:
t:The comparative form of bad is
:WORSE. Remember that it is an irregu-
t:lar form of comparison.
u:return

*ten
t:
t:Complete the following sentence:
t:
t:She is feeling a lot ----- (good) now.
t:
th:   ?
pr:u
a:
m:BETTER
uy:right
jy:once
u:wrong
t:
t:
t:The correct answer is BETTER.
u:return

*once
t:
t:True or False:
t:The superlative form of old can be
t:either OLDEST or ELDEST.
t:
th:    ?
pr:u
```

```
a:
m:TRUE
uy:right
jy:doce
u:wrong
t:
t:
t:The statement is true. Both forms are
t:correct. ELDEST is a little more
:formal.
u:return

*doce
t:
t:What is the superlative form of
t:MUCH and MANY?
t:
th:      ?
pr:u
a:
m:MOST!THE MOST
uy:right
ly:monsters1,score3
u:wrong
t:
t:
t:The correct answer is MOST or THE
:MOST as in:
t:This is THE MOST exciting
:trip I have ever taken.

u:return
l:monsters1,score3

*set4
c:m(4)=m(4)+1
c:c1=0
t:... This is set of questions #4 ...
t:
t:Combine the following sentences into
:one by using AS..AS:
t:
t:Mary is 22 years old.
t:Jane is 22 years old.
t:
th:?
pr:u
a:
m:MARY IS AS OLD AS JANE!JANE IS AS OLD
:AS MARY
uy:right
jy:trece
un:wrong
```

```
t:
t:The correct sentence would be:
t:
t:MARY IS AS OLD AS JANE. In this case
:we have a comparison in terms of
:equivalence of a quality.
u:return

*trece
t:
t:Combine these two sentences into one
:by using AS..AS:
t:
t:Math is interesting.
t:Calculus is interesting also.
t:
th:
pr:u
a:
m:MATH IS AS INTERESTING AS CALCULUS
uy:right
jy:ten4
un:wrong
t:
t:The correct sentence would be:
t:
t:MATH IS AS INTERESTING AS CALCULUS.
t:It is a comparison based on
:equivalence.

u:return
*ten4
t:
t:Which of the following sentences is
:not grammatically correct:
t:
t:1.Sometimes he behaves like a baby.
t:2.He works like a horse.
t:3.John looks like as his father.
t:4.John's book is like mary's.
t:
th:?
pr:u
a:
m:THREE!3
uy:right
jy:ten5
un:wrong
t:
t:Sentence number four is grammatically
:incorrect since it uses both LIKE and
:AS to express a comparison of
:equivalence, when only one is
```

```
:necessary.
t:
u:return
*ten5
t:
t:Which of the following sentences
:expresses a comparison of faulty
:logic:
t:
t:1.The novels he wrote are better than
:    the novels written by his contempo-
:    raries.
t:
t:2.The flowers in the spring are more
:    beautiful than the winter.
t:
t:3.He is more intelligent than his
:    sister.
t:
t:4.The electric typewriter is more
:    useful than the manual machine.
t:
th:?
pr:u
a:
m:2!TWO
uy:right
ly:monsters1,score4
un:wrong
t:
t:Sentence number 2 is not logically
:correct. It should be THE FLOWERS IN
:THE SPRING ARE MORE BEAUTIFUL THAN THE
:TREES IN THE WINTER. Or, THE SPRING IS
:MORE BEAUTIFUL THAN THE WINTER.
t:
u:return
l:monsters1,score4
e:
```

I-G Skedemo

```
r:program skedemo
g:es
g:m400,400


d:f$(20)
c:f$="muevelo"
s:50,10
d:m1$(167)
d:m2$(63)
d:m3$(166)
fix:3,f$

fi:1,m1$

fi:2,m2$

fi:3,m3$

tx:eskeleto

th:$m1$ $m1$ $m1$ $m1$ $m1$ $m1$ $m1$
:$m1$ $m1$ $m1$ $m1$ $m1$ $m1$ $m1$



th:$m2$

th:$m3$


tx:
gx:end
g:v0,39,20,23

w:5
```

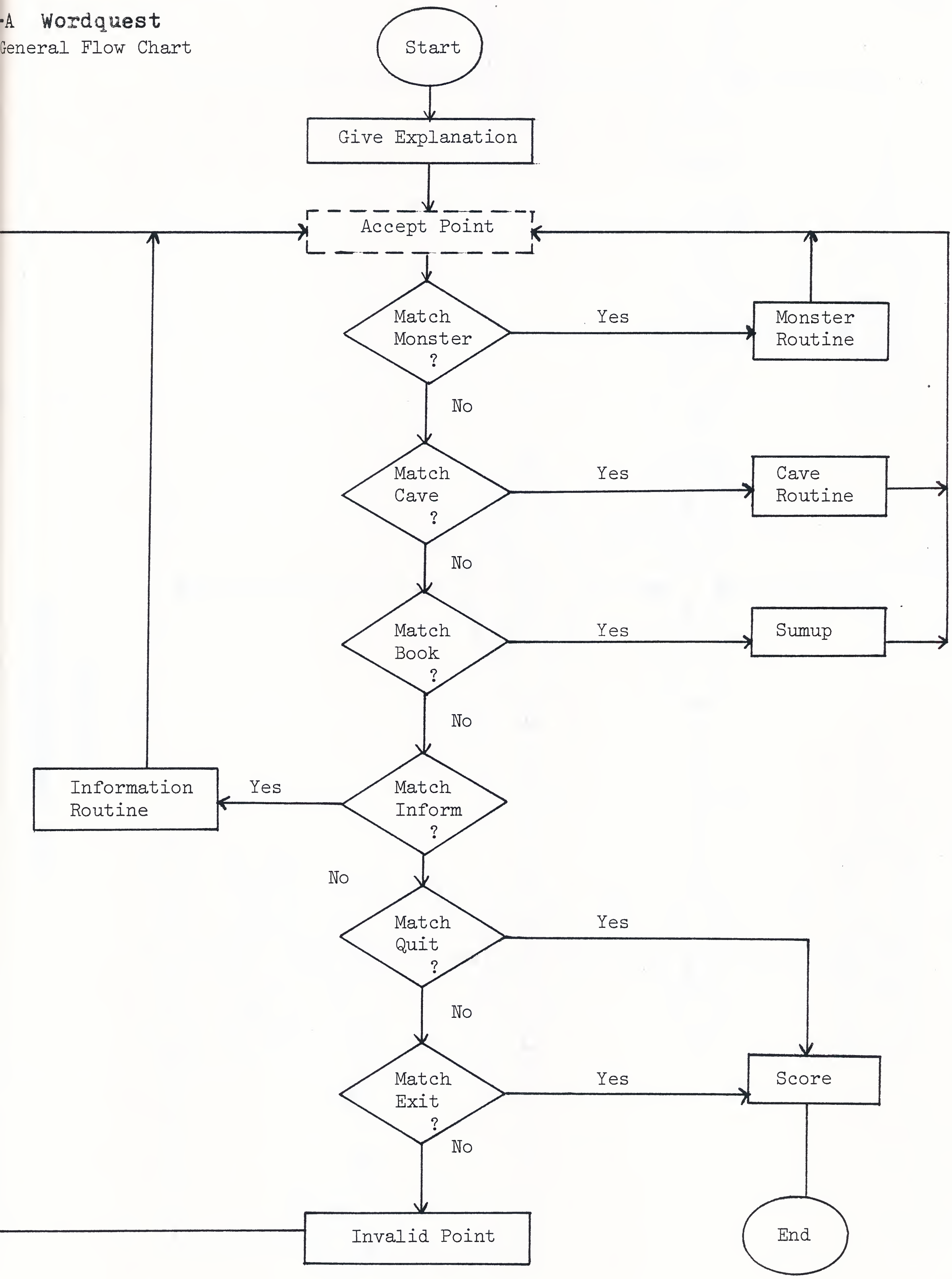
I-H Checkpoint

```
gx:graph1:stairs!
gx:graph1:stairscomp

*try

ap:
c:h=%x
c:v=%y
t:#h
w:4
t:#v
w:4
c:t=t+1
j:try
```

II. Flow Charts

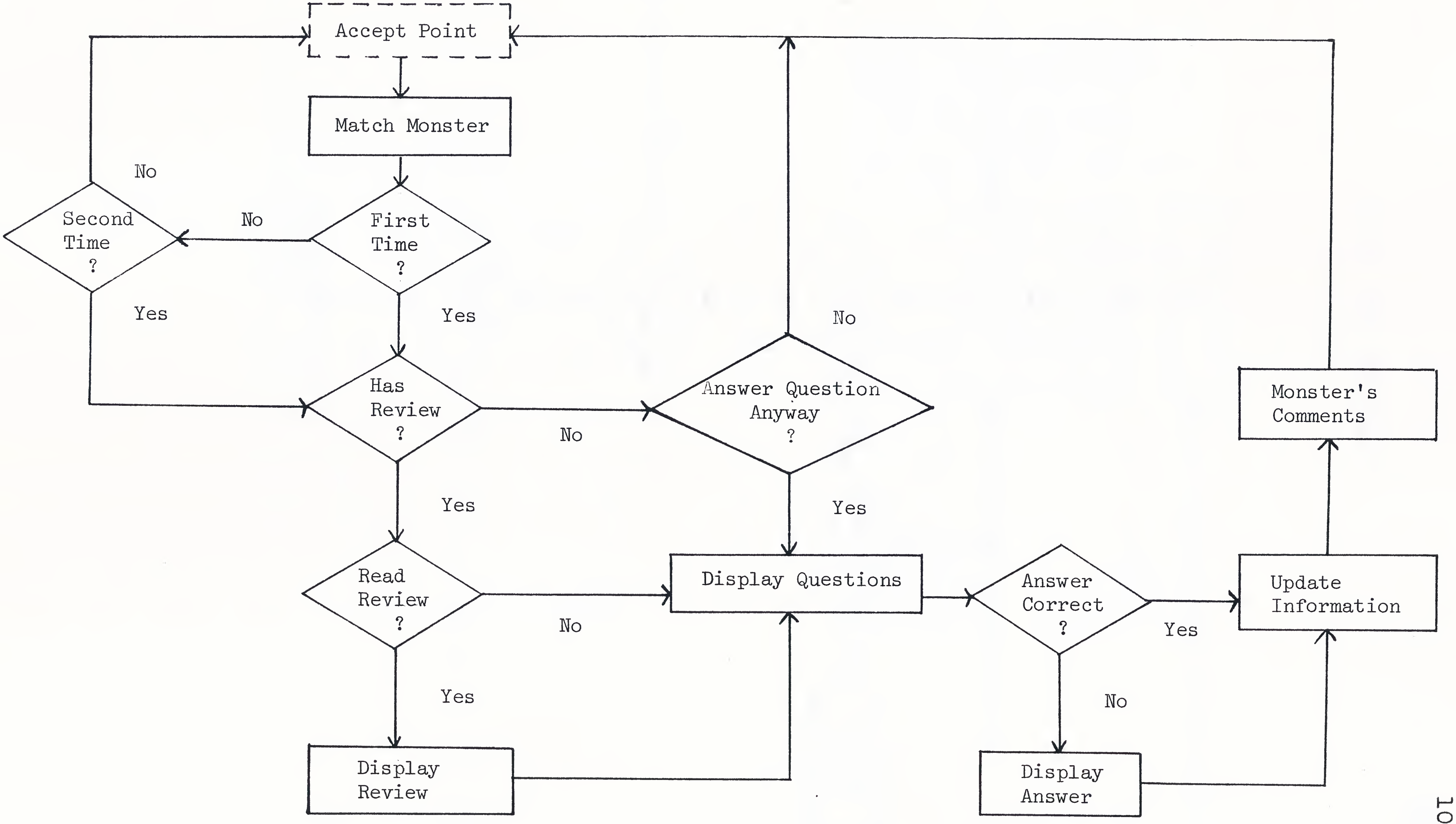
-A **Wordquest**

General Flow Chart



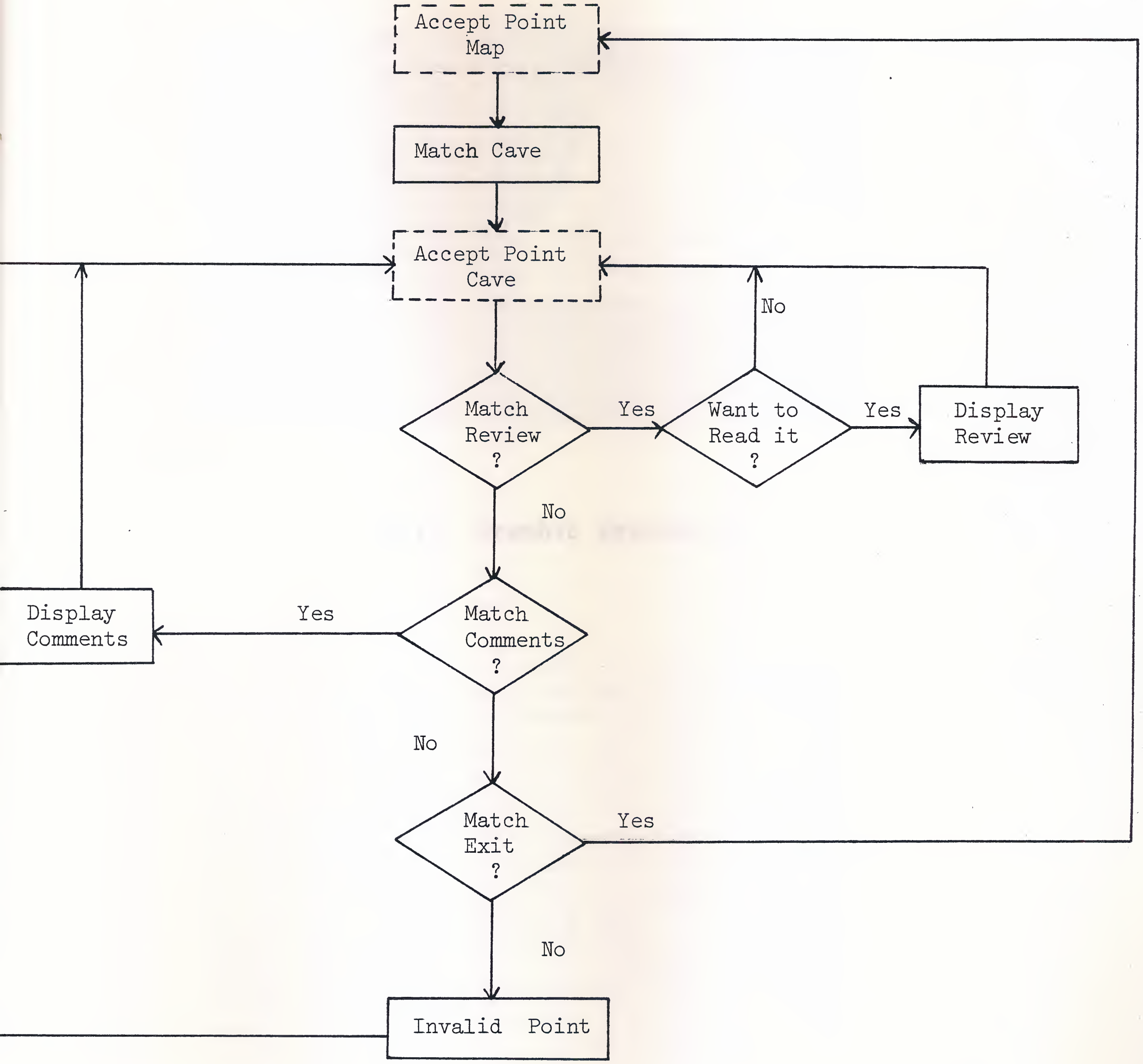
II-B Monster Routine

Flow Chart



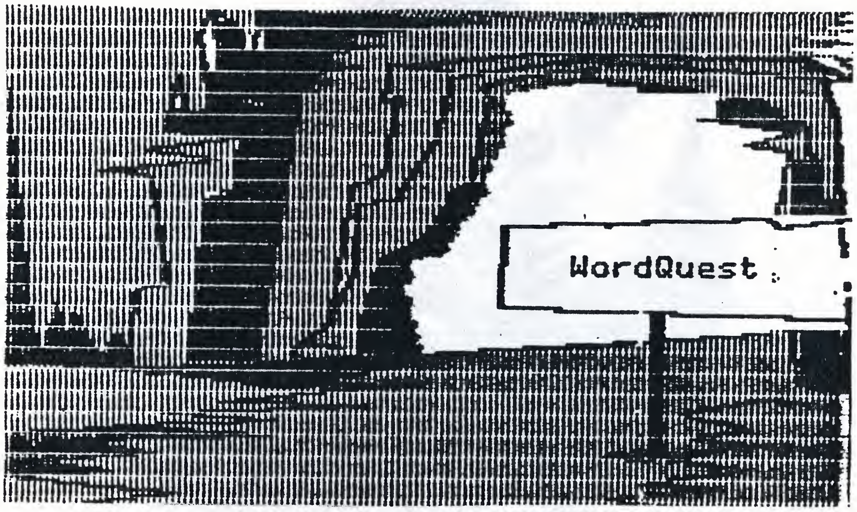
II-C Cave Routine

Flow Chart

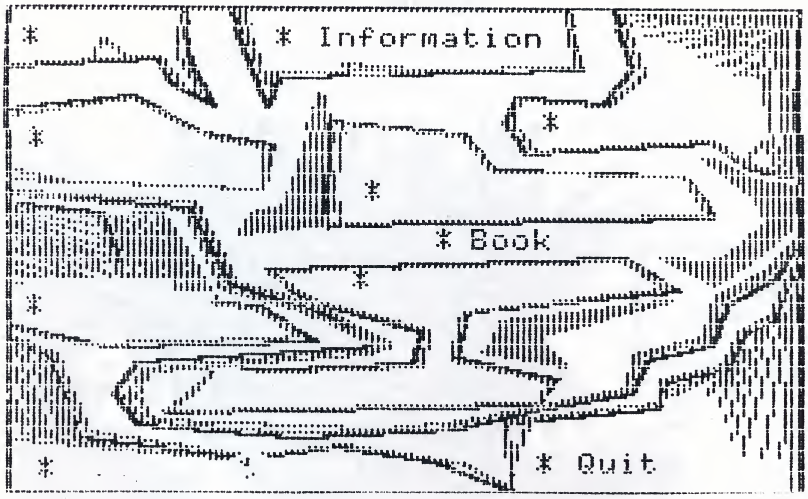


III. Graphic Printouts

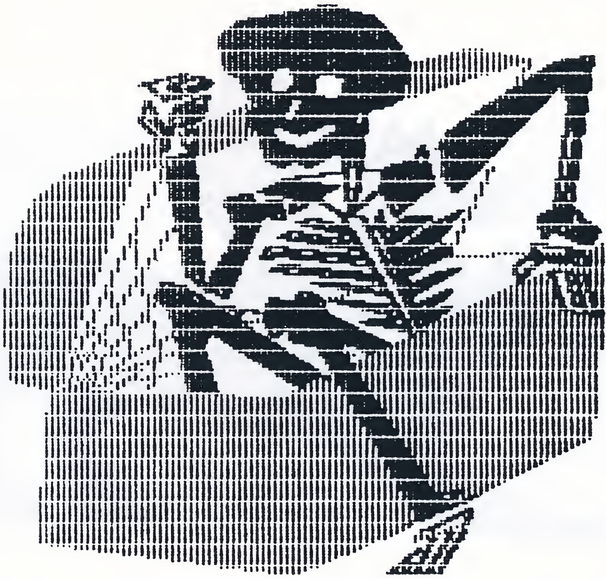
III.A CAVE ENTRANCE



III.B MAP 1

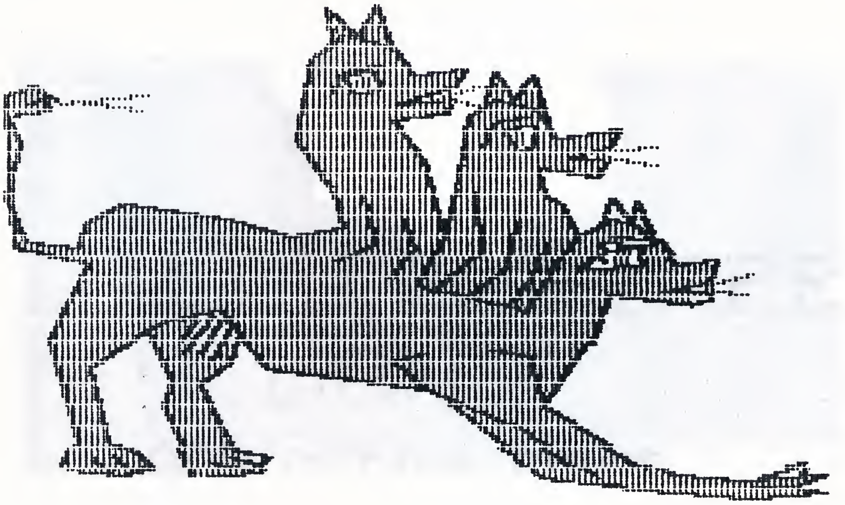


III.C ESKELETON

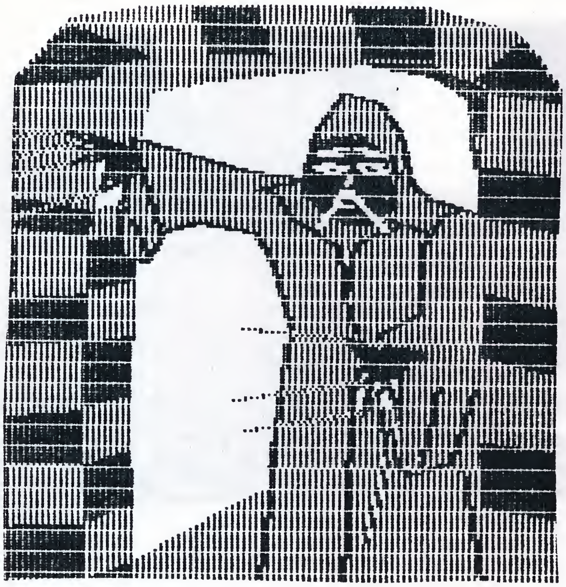


III.D DRAGONA

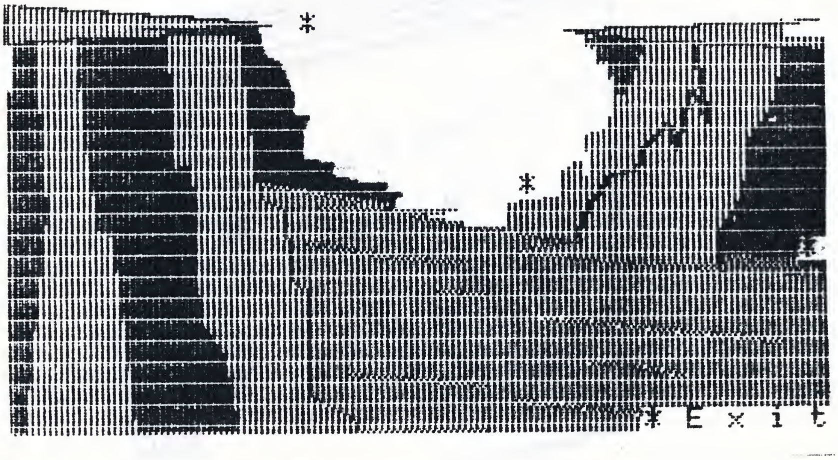
III.E CERBERUS



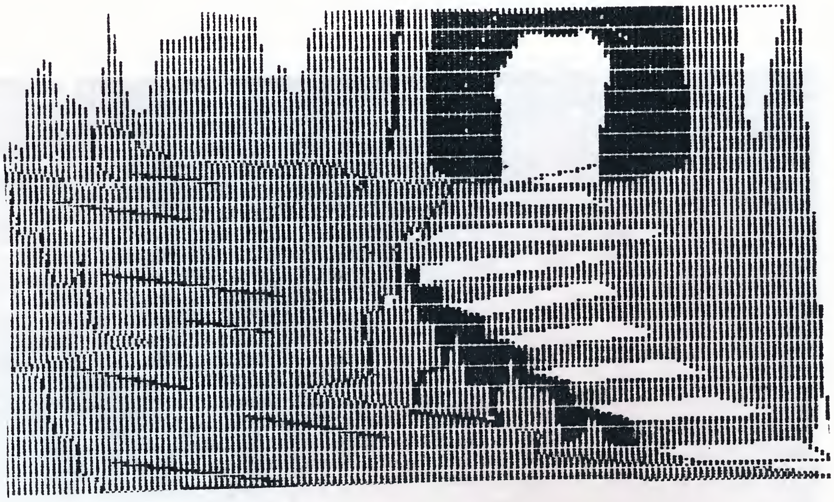
III.F WARLOCK



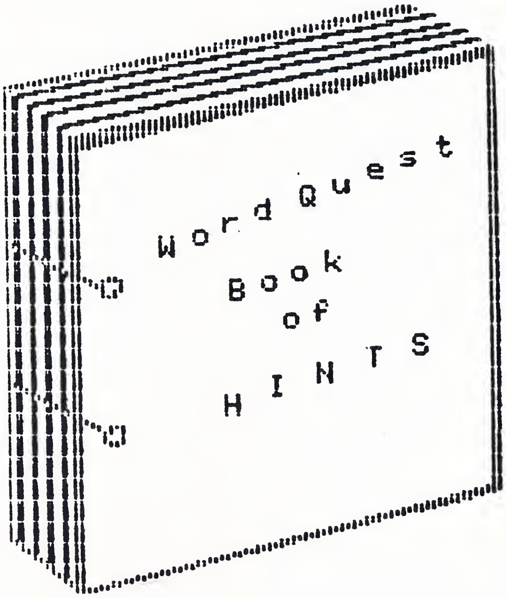
III.G CLIFF



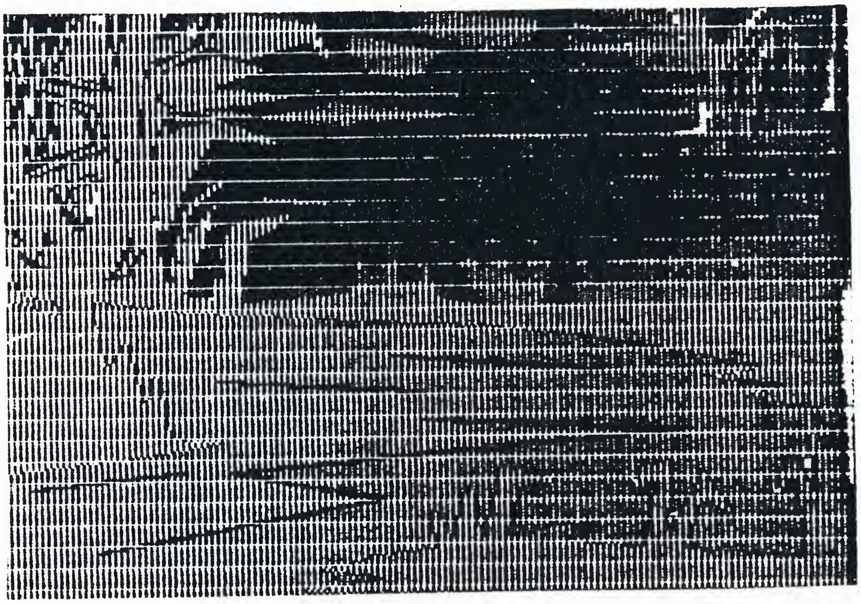
III.H RUINS



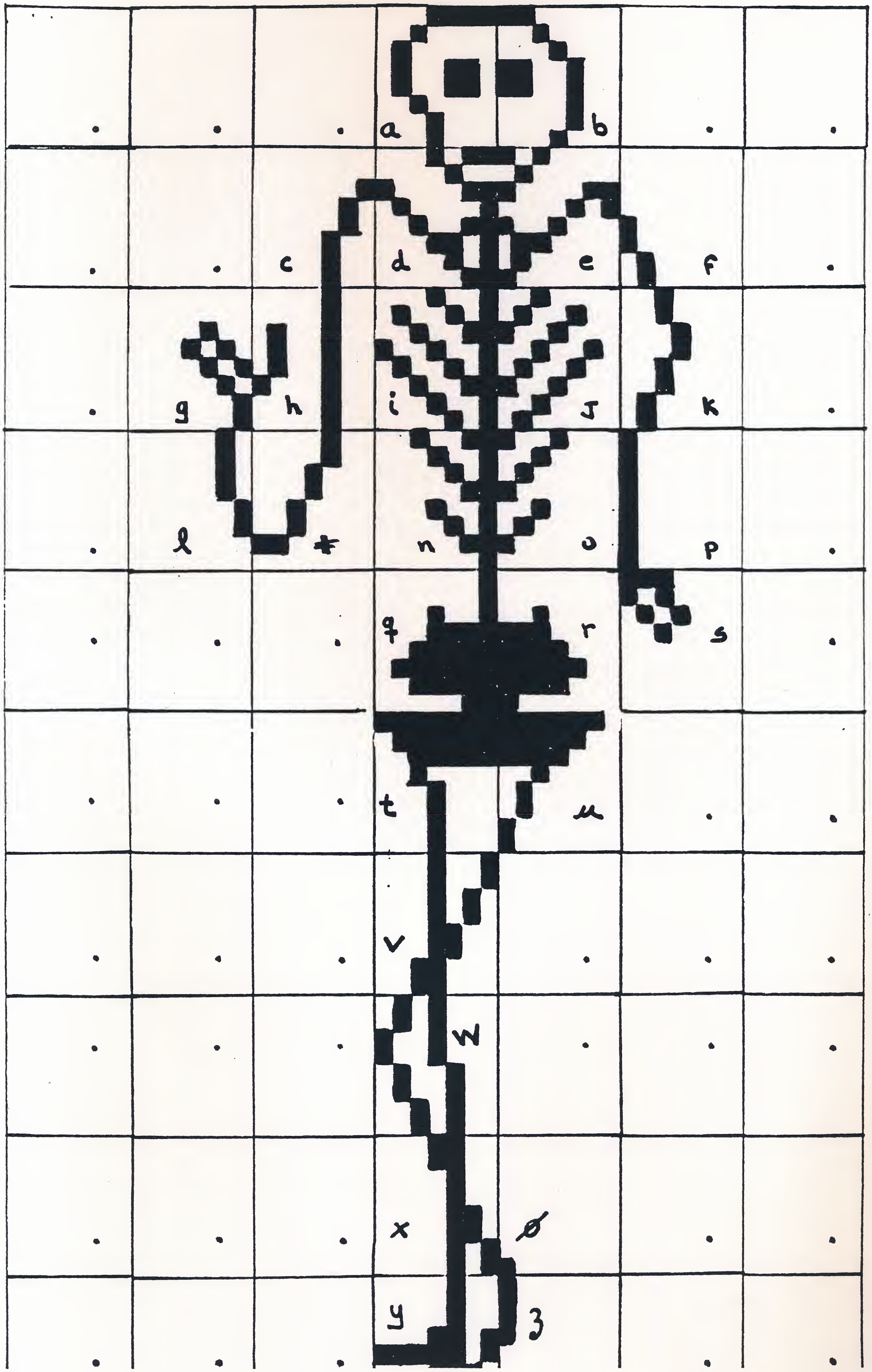
III.I BOOK



III.J CAVE EXIT



IV. Skeleton.
Sequence 1. Frame 1


a
b
c
d
e
f
g
h
i
J
K
l
*
n
o
p
q
r
s
t
u
v
w
x
ø
y
3